Coarse space for domain decomposition method with optimized transmission conditions

Ryadh Haferssas

Université Pierre et Marie Curie
INRIA

Doctoral School **École Doctorale Sciences Mathématiques de Paris Centre**

University Department **Laboratoire Jacques-Louis Lions**

Thesis defended by **Ryadh M. Haferssas**

Defended on **23rd November, 2016**

In order to become Doctor from Université Pierre et Marie Curie

Academic Field **Applied mathematics**

Espaces grossiers pour les méthodes de décomposition de domaine avec conditions d'interface optimisées

Thesis supervised by Frédéric Nataf

Committee members

Referees	Abderahmane Bendali	Professor at INSA de Toulouse
	Martin J. Gander	Professor at University of Geneva
Examiners	Bruno Després	Professor at University of Pierre et Marie Curie
	Victorita Dolean	Professor at University of Strathclyde
	Frédéric Hecht	Professor at University of Pierre et Marie Curie
	Pierre Jolivet	Junior Researcher at CNRS
	Christian Rey	Senior Researcher at Safran
Supervisor	Frédéric Nataf	Senior Researcher at University of Pierre et Marie Curie

Keywords: numerical analysis, domain decomposition methods, preconditioners, high performance computing, nearly-incompressible,geneo-2

This thesis has been prepared at the following research units.

Laboratoire Jacques-Louis Lions

4 place Jussieu
75005 Paris
France

☎ +33 1 44 27 42 98
Web Site http://ljll.math.upmc.fr/

INRIA

2 Rue Simone IFF,
75012 Paris
France

☎ +33 1 39 63 55 11
Web Site https://www.inria.fr

À ma famille

Le savoir est le premier des exils.

Malika Mokeddem

Le français… ma langue silencieuse.

Assia Djebar

Espaces grossiers pour les méthodes de décomposition de domaine avec conditions d'interface optimisées

Abstract

The objective of this thesis is to design an efficient domain decomposition method to solve solid and fluid mechanical problems, for this, Optimized Schwarz methods (OSM) are considered and revisited. The optimized Schwarz methods were introduced by P.L. Lions. They consist in improving the classical Schwarz method by replacing the Dirichlet interface conditions by a Robin interface conditions and can be applied to both overlapping and non overlapping subdomains. Robin conditions provide us an another way to optimize these methods for better convergence and more robustness when dealing with mechanical problem with almost incompressibility nature. In this thesis, a new theoretical framework is introduced which consists in providing an Additive Schwarz method type theory for optimized Schwarz methods, e.g. Lions' algorithm. We define an adaptive coarse space for which the convergence rate is guaranteed regardless of the regularity of the coefficients of the problem. Then we give a formulation of a two-level preconditioner for the proposed method. A broad spectrum of applications will be covered, such as incompressible linear elasticity, incompressible Stokes problems and unstationary Navier-Stokes problem. Numerical results on a large-scale parallel experiments with thousands of processes are provided. They clearly show the effectiveness and the robustness of the proposed approach.

Keywords: numerical analysis, domain decomposition methods, preconditioners, high performance computing, nearly-incompressible,geneo-2

Résumé

L'objectif de cette thèse est la conception, l'analyse et l'implémentation d'une méthode de décomposition de domaine efficiente pour des problèmes de la mécanique des solides et des fluides. Pour cela les méthodes de Schwarz optimisée (OSM) sont considérées et révisées. Les méthodes de décomposition de domaine de Schwarz optimisées ont été introduites par P.L. Lions, elles apportent une amélioration aux méthodes de Schwarz classiques en substituant les conditions d'interface de Dirichlet par des conditions de type Robin et cela pour les méthodes avec ou sans recouvrement. Les conditions de Robin offrent un très bon levier qui nous permet d'aller vers l'optimalité des méthodes de Schwarz ainsi que la conception d'une méthode de décomposition de domaine robuste pour des problèmes de mécanique complexes comportant une nature presque incompressible. Dans cette thèse un nouveau cadre mathématique est introduit qui consiste à munir les méthodes de Schwarz optimisées (e.g. L'algorithme de Lions) d'une théorie semblable à celle déjà existante pour des méthodes de Schwarz additives, on définit un espace grossier pour lequel le taux de convergence de la méthode à deux niveaux peut être prescrit, indépendamment des éventuelles hétérogénéités du problème traité. Une formulation sous forme de preconditioneur de la méthode à deux niveaux est proposée qui permettra la simulation parallèle d'un large spectre de problèmes mécanique, tel que le problème d'élasticité presque incompressible, le problème de Stokes incompressible ainsi que le problème instationnaire de Navier-Stokes. Des résultats numériques issues de simulations parallèles à grande échelle sur plusieurs milliers de processeurs sont présentés afin de montrer la robustesse de l'approche proposée.

Mots clés : analyse numérique, méthodes de décomposition de domaines, preconditioneurs, calcul haute performance, presque-incompressible, geneo-2

Laboratoire Jacques-Louis Lions
4 place Jussieu – 75005 Paris – France

Remerciements

Voici venu le moment de mettre des mots un peu moins cartésiens dans cette thèse. Avant toute chose, je souhaiterais remercier mon directeur de thèse Frédéric Nataf pour tout ce qu'il m'a apporté pendant ces trois années durant lesquelles j'ai bénéficié d'un encadrement d'une excellente qualité. Il est indéniable que la réussite de cette thèse doit beaucoup aux grandes qualités scientifiques de Frédéric, mais ses qualités humaines n'ont pas eu pour autant un moindre rôle, son optimisme, sa patience, ses encouragements constants m'auront permis de mieux faire face à la difficulté du travail. Très cher Frédéric, je te remercie infiniment pour tout !

Je voudrais remercier ici vivement Abderahmane Bendali et Martin J. Gander pour avoir accepté d'être les rapporteurs de cette thèse, de leur intérêt pour ce travail et des remarques constructives dont j'ai pu grandement bénéficiées.

Je tiens également à remercier Bruno Després, Victorita Dolean, Frédéric Hecht, Pierre Jolivet et Christian Rey. Je suis très honoré qu'ils aient accepté de faire partie de ce jury. Je profite de cette occasion pour remercier tout particulièrement Frédéric Hecht de son aide et de son précieux accompagnement, et cela depuis le début ma formation en Master à l'université Pierre et Marie Curie. J'ai eu la chance et le plaisir de suivre les cours de Frédéric aussi bien en Master1 qu'en Master2. Frédéric Hecht m'a définitivement introduit à l'univers du calcul scientifique, j'ai appris auprès de lui les rudiments et le b.a.-ba de la programmation, et bien d'autres choses puisque j'ai pu, à mon grand bonheur, bénéficier de son expertise dans le calcul scientifique tout au long de ma thèse, c'est dire à quel point j'ai eu de la chance.

Il y a une personne à qui je dois beaucoup et un paragraphe de remerciements ne suffirait pas pour la remercier à sa juste valeur. Je voudrais remercier chaleureusement Pierre Jolivet. Merci Pierre d'avoir été là à tous les moments de cette thèse. J'ai appris une quantité incroyable de choses auprès de Pierre, je retiendrai de lui, entre autres, le souci de la précision et la rigueur dans le travail bien fait. L'acronyme HPDDM apparaîtra à de multiples reprises dans cette thèse, ce n'est que la très belle œuvre de Pierre sur laquelle j'ai eu le privilège travailler et de pouvoir y apporter une maigre contribution.

Je souhaiterais également remercier Nicole Spillane, ma sœur spirituelle de thèse. J'ai eu la chance de commencer ma thèse alors qu'elle terminait la sienne, ce qui m'a permis de prendre le train déjà bien en marche dans les meilleures conditions qui soient. Merci Nicole pour tout le temps que tu m'a accordé, tes conseils, tes encouragements et ton partage d'expérience ont été précieux.

Tout au long de cette thèse, j'ai eu la chance d'être plongé dans deux environnements en même temps. D'abord le laboratoire Jacques-Louis Lions, un endroit très agréable, je remercie tous les membres du laboratoire qui en font un lieu où règne une très bonne ambiance, mais aussi un lieu d'une très grande qualité scientifique. J'ai pu y rencontrer des personnes brillantes et d'une modestie sans égale, je remercie tout particulièrement Sidi-Mahmoud Kaber qui m'a donné l'occasion d'enseigner à ses cotés, il se trouve que j'ai été d'abord son étudiant avant de devenir son collègue. Je remercie également Mme Edwige Godlewski pour sa bienveillance envers ses étudiants, j'ai eu le plaisir d'en faire partie et je garderai toujours en mémoire les moments de discussions et les riches conseils qu'elle m'a accordés et cela bien avant le début de la thèse. Je remercie l'ensemble des assistantes administratives Salima, Malika, Nadine et Catherine qui m'ont accompagné à de nombreuses reprises dans mes démarches annuelles pour obtenir le fameux titre de séjour.

Je remercie les gens de l'équipe Alpine de l'INRIA, à leur tête Laura Grigori qui ma permis de faire partie du projet Alpines, j'ai bénéficié tout au long de cette thèse de beaucoup de moyens.

Je n'oublie pas de remercier particulièrement Laurence qui a toujours été d'une efficacité extraordinaire, Xavier, Pierre-Henry, le collègue par excellence, je suis toujours impressionné par sa capacité à comprendre les problèmes des autres, son aide à de multiples reprises a été salutaire. Je remercie tous les doctorants et postdocs qui font d'Alpines une équipe particulièrement excellente, Simplice, Ange, Sébastien, Hussam, Olivier et Alan.

Beaucoup de membres du laboratoire ont été mes professeurs avant de devenir mes collègues, Corentin Audiard, Muriel Boulakia, Emmanuel Trélat, Pascal Frey, Jean-Frédéric Gerbeau, Marie Postel, Didier Smets. Un remerciement particulier pour Cristinel Mardare qui m'a fait découvrir un beau résultat de Newton-Kntorovitch lors d'un petit stage en M1. Merci Cristinel pour tout le temps que tu m'a consacré, j'ai énormément appris de toi, ta façon de jouer avec les équations dans le plaisir et la rigueur n'a pas manqué de me faire découvrir une toute autre facette des mathématiques.

Je souhaiterais remercier mes nombreux collègues de bureau, tout d'abord ceux du bureau 15-16-303 Lucile, Cécile, Geneviève, Nora, Stamatina, May et Sophie un bureau exclusivement féminin, dans lequel je n'ai été qu'une erreur de mesure à mon grand bonheur. Ensuite mes collègues du bureau 15-25-324, Clément, Frédéric, Hussam, Shengquan, Mireille, plus récemment Pierre, Karim, Jean à qui je souhaite une agréable aventure avec leur début de thèse. Comment ne pas évoquer Mehdi un collègue, un ami et même plus que cela un camarade d'armes de tous les temps. Merci Mehdi d'avoir été présent durant toutes ces années. Tes nombreuses qualités humaines sont rares.

Je n'oublie pas évidemment tous les gens qui continuent de m'accompagner depuis bien longtemps, je commencerai par les membres ma famille. En premier par mon oncle Nordine, merci Nounou pour beaucoup de choses, merci d'avoir été présent à tous les moments de ma scolarité et à celle de tous mes autres cousins, tout ce monde a eu le privilège d'accéder à des études supérieures et cela ne se serait pas fait sans ton accompagnement et ta bienveillance. Merci de nous avoir inculqué les besoins de la bonne instruction et de la culture. Merci à tous mes oncles et tantes, cousins, cousines, ô combien nombreux.

Durant un parcours scolaire, nous sommes forcément influencés par des modèles, j'ai eu la chance de croiser les bons, je pense tout particulièrement à Madame Targhit que je remercie infiniment, elle nous a inoculé le virus des belles lettres et cela n'a pas de prix.

Bien en dehors des mathématiques mais pas si loin, j'ai eu le plaisir de connaître des personnes sous d'autre cieux tout aussi raffinés. Je tiens à remercier tous mes amis Rythmeharmoniens de musique andalouse, à leur tête Beihdja Rahal, un nom qui n'est pas des moindres, "la diva" de la musique andalouse (elle n'aime pas qu'on l'a surnomme ainsi et pourtant elle l'est indéniablement) et puis Nacéra, Malika, Nawel, Meriem, Khaled, Mohamed, Dihia, Asmahane, Samia, Lina, Sabrina, Mohamed, Manel. Tout ce beau monde m'a permis de sortir de l'univers de mon boulot pour passer à chaque fois un agréable moment musical.

Je remercie tous les amis d'Alger et de Paris qui ont été parfois plus impatients que moi de voir la fin de cette thèse, Houcine, Réda, Sabrina, Fouad, Nina, Atika, Waheb Nourdine, Anouar, Abdou. Je ne compte plus les années qui font ces amitiés.

Mes derniers remerciements vont á ma famille, Ma frangine Djazia et le petit frangin Sofiane, mes parents sans qui rien de tout ceci n'aurait été possible. Leur soutient indéfectible, leur accompagnement, leur affections, leur souci de notre bonne instruction m'auront permis d'être ce que je suis et là où je suis aujourd'hui. Je tiens à leur dire que ce travail n'est que le fruit de tous leurs sacrifice qui a été infiniment grand durant toutes ces années.

Contents

List of Figures

Chapter 1

Introduction

Contents

1.1 Introduction : version française

1.1.1 Contexte de la thèse

La modélisation mathématique des phénomènes mécaniques est devenue une nécessité quasi absolue si on veut mieux comprendre et prédire leurs évolutions. Cependant, la résolution par des méthodes numériques des équations aux dérivées partielles (EDP) correspondantes donne lieu a des systèmes linaires algébriques qu'on doit résoudre. Tout modèle a pour ambition de refléter la réalité d'un quelconque phénomène de la manière la plus proche possible. Ainsi plus le modèle est riche plus les systèmes linéaires résultants sont gigantesques en terme de taille.

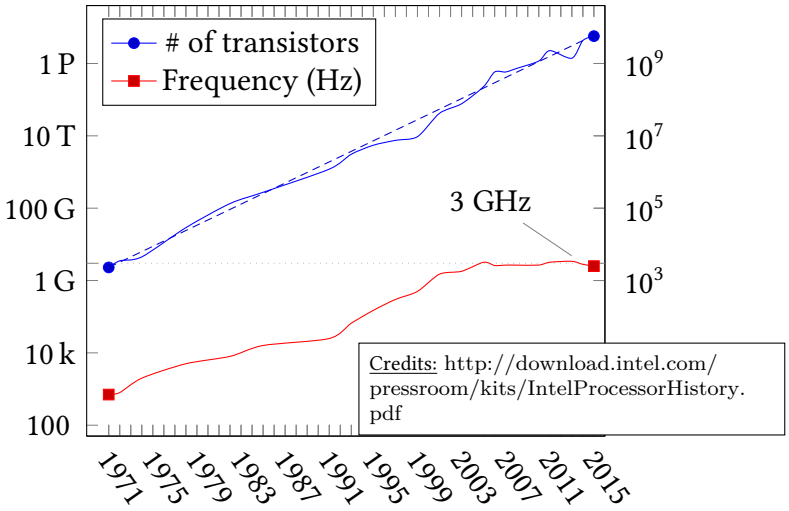


Fig 1.1 – Frequency and number of transistors of some Intel processors clearly showing the end of frequency scaling around the 3 GHz mark(by courtesy of P. Jolivet) .

Jusqu'à la fin des années 1990 et même quelques années après, les algorithmes des simulations numériques gageaient fortement leurs performances sur les performances des nouveaux processeurs qui allaient apparaitre dans le futures proche. Cette situation n'était pas si contraignante car la certitude apportée par la loi de Moore prédisait une amélioration des performances en calcul des processeurs. Il suffisait d'attendre la prochaine génération de processeurs pour voir que les performances de ces mêmes algorithmes de simulation numérique s'améliorer d'une manière significative. Hélas, cet âge d'or a connu une fin brutale comme peut l'attester la figure 1.1. Depuis l'année 2004, à cause des limites physiques, la fréquence d'horloge (baromètre de la puissance de calcul) des processeurs a cessé d'évoluer, au moment même où la complexité des nouveaux algorithmes de simulations devenait de plus en plus importante grâce aux différentes avancées dans l'enrichissement des modèles numériques. Cela a conduit la communauté du calcul scientifique à vouloir repenser fondamentalement le processus des simulations. C'est ainsi qu'avec l'apparition des architectures parallèles, le parallélisme des algorithmes est devenu un standard de conception. Ce mode de pensée est devenu une nécessité dans toute la chaine de la modélisation numérique, et cela de la mise en équation jusqu'à la résolution des systèmes linaires engendrés.

Dans le processus d'une simulation numérique, le temps passé dans la résolution des systèmes linéaires reste à ce jour significativement conséquent et même le plus important. Le principal intérêt qui a conduit l'entreprise de la présente thèse réside dans l'amélioration des moyens de résolution des systèmes linéaires de grande taille pour des calculateurs à architectures parallèles. Traditionnellement, il existe deux types d'approches pour résoudre les systèmes linéaires de grande taille résultants des schémas de discrétisation des EDP: les méthodes directes et itératives. Les méthodes directes sont connues pour être extrêmement robustes, cependant elles sont aussi très gourmandes en mémoire. Les exigences en place mémoire deviennent de plus en plus importantes lors de la résolution parallèle des grands systèmes linéaires. Par ailleurs, les méthodes itératives n'ont pas besoin d'autant d'espace mémoire mais elles présentent l'inconvénient d'être très peu robustes vis-à-vis des problèmes complexes. Les méthodes de décomposition de domaine ainsi que les méthodes multigrilles présentent l'avantage d'utiliser les deux approches traditionnelles dans une même procédure en ne préservant que le meilleur de chacune.

1.1.2 Méthodes de décomposition de domaine

Les méthodes de décomposition de domaine sont considérées comme des méthodes relevant du paradigme " diviser pour régner". Ce sont des méthodes qui proposent de diviser le domaine de calcul global en une multitude de petits sous domaines sur lesquels on résout des problèmes locaux de tailles relativement petites en prenant en compte les conditions d'interface adéquates. C'est dans cet aspect de localité que réside le parallélisme naturel des méthodes de décomposition de domaine. Par ailleurs, même si l'apparition de ce type de méthode remonte a plusieurs décennies, le véritable regain d'intérêt porté pour les méthodes de décomposition de domaine n'a été observé au sein de la communauté du calcul scientifique qu'avec l'avènement des architectures parallèles dans les calculateurs. Des améliorations significatives et continues voient le jour.
Les méthodes décomposition de domaine peuvent être réparties en deux grandes familles, les méthodes sans recouvrement, qui donneront lieu aux méthodes appelées méthodes de sous-structuration et celles avec recouvrement dites méthodes de Schwarz. Du fait de leur parallélisme naturel, elles sont considérées comme étant hautement concurrentes et suffisamment robustes lorsqu'elles sont utilisées sur des problèmes complexes. Dans le cadre de la présente thèse il sera essentiellement question des méthodes avec recouvrement. Des efforts d'amélioration seront apportés aux méthodes de Schwarz. En effet une variante optimisée robuste pour des problèmes à la fois de mécanique des solides et des fluides sera développé tout au long de ce travail.

1.1.3 Résumé & Contributions

L'objet principal de cette thèse est la conception, l'analyse et l'implémentation d'une méthode de décomposition de domaine robuste afin de traiter le plus large spectre possible de problèmes mécaniques essentiellement décrits par un système d'EDP. Dés lors, on se fixe comme principal objectif de proposer une méthode efficace dans le sens où elle doit être parfaitement extensible lors des simulations parallèles à grande échelle.
Afin d'atteindre ce but, notre stratégie développée tout au long de cette thèse sera guidée par les deux points suivants

- La robustesse de l'algorithme doit être garantie. Pour cela la méthode est immergée dans un cadre mathématique adéquat. Ce dernier nous mènera vers un résultat de convergence certaine pour des matrices symétriques définies positives générales, indépendamment des éventuelles hétérogénéités dans les coefficients.

- La construction des méthodes doit être la plus algébrique possible. Cela rendra d'autant plus faisable l'implémentation dans un code de méthode de décomposition de domaine existant. En effet, cela sera le cas au sein de la librairie HPDDM [JHNP13; JN14].

En vue d'aller encore plus loin, une extension sur le plan numérique sera faite pour un autre type de difficulté et ce pour des systèmes d'EDP comprenant la nature incompressible du problème traité, pour des systèmes d'EDP donnant lieu à des problèmes sous la forme point-selle ainsi que des systèmes d'EDP ayant une formulation non symétrique. Tout cela dans le contexte des simulations numériques parallèles à grande échelle des modèles mécaniques. On notera que le travail conduit lors de cette thèse reposera sur l'amélioration en continue des méthodes de Schwarz avec recouvrement.

Le présent manuscrit est structuré autour des quatre chapitres suivants.

Dans le chapitre 2, un rappel est proposé au lecteur sur les Méthodes de Décomposition de Domaine (MDD). Néanmoins dans cette thèse, on se focalisera essentiellement sur les méthodes de type Schwarz dites avec recouvrement. Ainsi, elles feront l'objet de quelques améliorations tout au long des chapitres suivants. Dans une première partie de ce chapitre 1

- Une introduction historique est donnée sur les méthodes de décomposition de Schwarz ainsi que l'algorithme itératif sous-jacent.
- Une version parallèle continue de l'algorithme de Schwarz est présentée.

Dans la seconde partie, on montrera comment les algorithmes de décomposition de Schwarz ont fait l'objet d'une reformulation dans le but de les adapter au contexte du calcul parallèle haute performance. Pour ce faire, des ingrédients nécessaires sont introduits afin de faciliter une traduction purement algébrique et moderne des méthodes de décomposition de domaines. On verra dans ce même chapitre que cela conduira naturellement à la construction de "preconditioneurs" à base de méthodes de décomposition de domaines.

Dans le chapitre 3, Une analyse des méthodes de Schwarz avec recouvrement est faite. L'objectif principal de ce chapitre réside dans l'étude du comportement de la convergence des algorithmes de Schwarz. En effet dans un travail préliminaire, nous rappellerons une stratégie pour faire ressortir et cela d'une manière analytique le taux de convergence pour une méthode de Schwarz sur un type de problème d'EDP donné. À ce niveau, afin d'aller vers une amélioration possible de la convergence globale, une compréhension profonde de l'influence des conditions de transmission aux interfaces devient nécessaire. Ce qui va nous conduire à agir en conséquence. On notera que l'objectif visé est d'appliquer efficacement les méthodes de Schwarz aux équations d'élasticité dans le cas de la limite incompressible. Ce travail sera étendu par la suite aux équations de Stokes incompressible.

- On place le problème d'EDP dans un cadre à deux dimensions dans le but de simplifier l'écriture du système d'équations. Le système ainsi simplifié sera ensuite exprimé dans l'espace de Fourier. Cela réduira la dimension d'espace et nous permettra l'expression analytique des solutions du problème originaire de l'EDP de départ.
- Le type des conditions de transmission aux interfaces a une influence certaine et importante sur le comportement du taux de convergence. A travers ce travail, on va s'en convaincre en mettant en oeuvre et en comparant plusieurs types de conditions de transmission dans le cas du système d'élasticité par exemple, les conditions de Dirichlet, conditions de Neumann, un mixte Neumanm-Dirichlet, Une observation majeure est faite sur la très mauvaise convergence pour Dirichlet et Neumann lorsqu'on traite la limite presque incompressible du problème élastique.
- Un autre type de conditions de transmission aux interfaces est présenté, connu comme condition de Robin (Fourier). Cela apparait comme le meilleur moyen d'offrir une réelle possibilité d'aller vers l'optimalité de la convergence de la méthode de Schwarz.
- Grâce aux conditions de type Robin, une approximation de l'opérateur DtN (Dirichlet-to-Neumann) est construite d'une manière purement algébrique.
- Ainsi une comparaison entre tous les types de conditions de transmission aux interfaces évoqués précédemment est faite dans le cadre d'un problème d'élasticité d'abord pour un

matériau compressible, ensuite pour un matériau presque incompressible. Il apparaîtra que l'utilisation des conditions optimisées via l'opérateur DtN donne une amélioration significative du taux de convergence pour une méthode de décomposition de domaine de type Schwarz.

Les constructions algébriques ainsi que les résultats obtenus dans ce chapitre sont fournis sous forme de code Maple dans l'annexe A.

Dans le chapitre 4, l'algorithme de P.L. Lions est introduit. Apparu comme une amélioration de l'algorithme de Schwarz à travers l'usage des conditions aux interfaces de type Robin développées lors du chapitre 2, le principal objectif de ce chapitre consistera à munir l'algorithme de Lions d'une théorie semblable à celle déjà existante depuis plusieurs années pour les algorithmes de décomposition de de domaine type Schwarz additif. Il s'ensuivra un focus sur les applications aux problèmes d'élasticités dans le cas de la limite presque incompressible. Pour ce faire, les étapes suivantes sont élaborées

- Un rappel est fait sur la version de l'algorithme de P.L. Lions ainsi que celui de son équivalent algébrique plus connu sous l'acronyme ORAS (Optimized Restricted Additive Schwarz).

- Une variante symétrique de ORAS est introduite, baptisée avec l'acronyme SORAS (Symmetric Optimized Restricted Additive Schwarz).

- Un cadre mathématique sera considéré. Dans cette partie nous verrons que le cadre du lemme des espaces fictifs se présente comme l'environnement le plus adéquat.

- On construit un espace grossier à base de conditions optimisées afin d'équiper l'algorithme de P.L. Lions d'un second niveau. Cela donnera lieu à la méthode désormais dénommée (SORAS-GenEO2) avec un taux de convergence prescrit théoriquement à l'avance.

- On montrera à travers quelques points que la méthode développée est tout-à-fait applicable numériquement aux problèmes type point-selle sans aucun effort supplémentaire.

- En dernier lieu et afin de montrer la bonne robustesse de la méthode de décomposition de domaine de Schwarz optimisée à deux niveaux développée précédemment (i.e. SORAS-GenEO2), des tests d'extensibilité sont effectués. D'abord un test d'extensibilité faible est réalisé avec un problème d'élasticité sur un matériau hautement hétérogène comportant à la fois des parties compressibles et presque incompressibles. En second lieu, un test d'extensibilité forte est achevé avec succès sur un problème de Stokes incompressible avec une extensibilité allant jusqu'à 8192 processeurs.

Dans le chapitre 5, le principal but est de traiter le système instationnaire non linéaire des équations de Navier-Stokes de la façon la plus efficace possible. On verra comment appliquer la méthode de décomposition de domaine de type Schwarz optimisée développée au long des précédents chapitres. Le chapitre est structuré sur les points suivants

- Une présentation du systèmes des équations de Navier-Stokes est écrite dans le cadre du benchmark des écoulements de Turek [STDKR96].

- Un rappel est fait sur une méthode de déflation. Sachant que la simulation d'un problème instationnaire requière la résolution de plusieurs systèmes linéaires, une des manières de tirer avantage des résolutions successives est d'intégrer une méthode de recyclage de de l'espace de Krylov engendré à chaque construction pour pouvoir ainsi l'utiliser par le bais des techniques de déflation lors de la prochaine résolution. Cela permet une réduction significative du nombre d'itérations de la méthode itérative utilisée. Les préconditioneurs ORAS et SORAS seront utilisés conjointement avec une méthode de recyclage et de déflation .

- Dans le but de montrer le bon comportement des algorithmes de Schwarz optimisés, on procède à une validation d'un test de Turek dans le contexte d'une simulation parallèle à grande-échelle. On observera une extensibilité optimale à travers des tests d'extensibilité jusqu'a 16384 processeurs

1.2 Introduction : english version

1.2.1 Context of the thesis

Mathematical modeling of mechanical phenomena has become a necessary process to better understand and predict their evolution. However the numerical resolution of the corresponding partial differential equations (PDE) leads to linear algebraic system to be solved, the higher the ambition of the model to be accurate, the largest is the resulting linear system to solve.

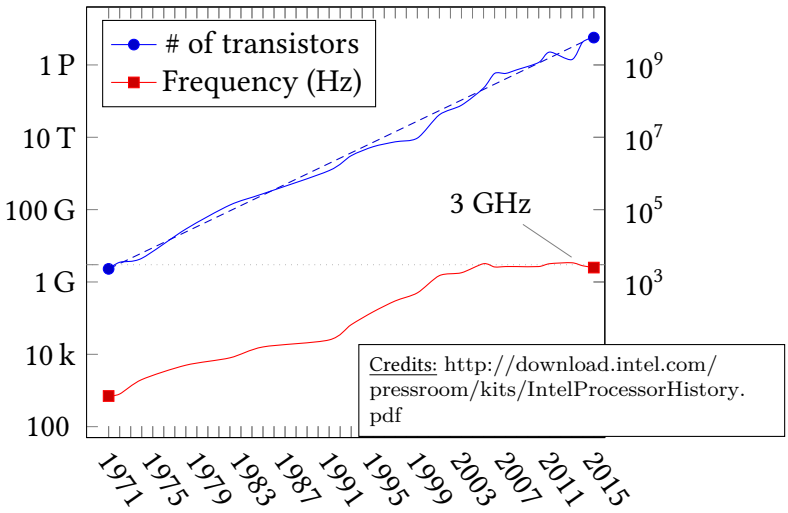


Fig 1.2 – Frequency and number of transistors of some Intel processors clearly showing the end of frequency scaling around the 3 GHz mark(by courtesy of P. Jolivet) .

Until the end of 1990s and even few years later, the numerical simulation algorithms were strongly dependent in the improvement of a single computer performance. On the other side it was not disturbing, with the Moore's law and its relation with the frequency of the chips, single processors were evolving at an impressive rate. Just waiting for the next generation of processors guaranteed a better performance. Unfortunately this golden age had an abrupt end, as we can show in figure 1.2 du to a physical limit (heat dissipation) the Moore's law is no longer true since 2004. But complexity of the simulation algorithms and particularly the size of linear system does not stagnate. Computer architectures evolve to a parallel paradigm which becomes a reference model. The scientific computing community turned naturally to parallel computing. Thinking "parallel" is by now necessary to design new methods in order to solve linear systems resulting from the PDE modeling.

In model simulation process, among all the parts, the solve of the linear systems still takes the major part in the whole time consumption. The main interest which has driven our work in this thesis is the efficiency in solving large algebraic linear system in parallel experiments. Traditionally when we have to solve a large linear system arising from a discretization of PDE on a parallel architecture, two types of approaches are used : direct and iterative solvers. Direct solvers are known to be very robust, however the memory requirement becomes significant with larger systems. On the other hand iterative solvers are less memory consuming and naturally parallel but they suffer from a lack of robustness. Domain Decomposition method as well as multigrid method are hybrid methods that take advantage of direct solvers and iterative

solvers in the same algorithm.

1.2.2 Domain Decomposition Methods

Domain decomposition methods can be seen as a ”divide and conquer” technic. They propose to divide the computational domain in many subdomains on which we solve local problems with the adequate interface conditions. This is a very natural parallel paradigm to solve the PDE problem. Even if domain decomposition method have been known for a long, with the expansion of parallel computer architecture they received a renewed interest from scientific computing community which induced an important boost in their improvement. Domain decomposition method can be divided into two larges families, the non overlappings methods, so-called sub-structuring methods and overlapping methods such as Schwarz method. Because of their natural parallelism, both are highly concurrent methods. They are robust enough to solve a large spectrum of complex problems, In this thesis we will be interested in the improvement of the Schwarz methods. We will propose a robust optimized version when dealing with both fluid and solid mechanical problems.

1.2.3 Summary & Contributions

The main subject of this thesis is the design and the analysis of a robust domain decomposition method in order to tackle the largest range of mechanical resulting PDEs problem. We keep in mind that the main objective is to get the best scalable efficiency when running parallel large scale experiments.
To achieve this goal, our developed strategy in this thesis relies on the following points

- Robustness of the algorithm must be guaranteed. For this, the method is expressed in an appropriate mathematical framework. The latter will guarantee convergence results for general symmetric positive definite (SPD) problems regardless of the high heterogeneity in the coefficients.

- The construction must be at the most algebraic level as possible. It will be ready to be implemented in an existing code for domain decomposition method. This will be the case for HPDDM [JHNP13; JN14] library.

An extension is made for another range of difficulties, such as the PDE problems incorporating incompressibility features, saddle point problem and even with a non symmetric problem in the context of large scale parallel simulation of mechanical modeling problem. To achieve this goal, the strategy developed in this thesis relies on an improvement of overlapping Schwarz algorithm.

The manuscript is structured in four chapters.

In chapter 1, we recall to the reader some Domain Decomposition Methods (DDM) existing in the literature, with a particular focus on those with overlaps, the so-called Schwarz methods, which will be the main subject to be enriched and improved in the subsequent chapters. In the first part

- Historical introduction is given of Schwarz domain decomposition and the resulting iterative algorithm.

- Parallel continuous version of the Schwarz algorithm is presented.

In the second part of this chapter, we have the aim to present how the Schwarz decomposition algorithm was reformulated in order to be well fitted for performing domain decomposition algorithm in computational framework. The necessary ingredients are introduced in order to translate Schwarz domain decomposition to modern algebraic formulation. This leads to the construction of preconditioners based on Schwarz Domain Decomposition.

In chapter 2, the analysis of overlapping Schwarz method is presented. The main purpose is the study of the convergence rate, then find the best possible way to improve it. To achieve this, it requires a deep understanding of the influence of transmission conditon on the convergence behavior. Note that the targeted objective is to apply the Schwarz DDM to elasticity equations in the incompressible limit. This work was easily extended to Stokes equations.

- We place the PDE problem in a simple two dimensional framework in order to simplify the writing of the equations, then express the problem in Fourier space. This will reduce the dimension space and allows for a analytic solution of the equations.
- The type of transmission conditions between subdomains in Schwarz algorithm has an important influence on the convergence behavior of the domain decomposition algorithm. This leads us to explore throughout this work the variety of interface transmission conditions as a local boundary conditions in the case of Elasticity equations, Dirichlet conditions, Neumann conditions, a mix between Neumann and Dirichlet. The main observation is made on the bad convergence behavior for Dirichlet and Neumann when dealing with nearly incompressible materials.
- An other type of transmission conditon is introduced, so-called Robin condition, at this stage the use of Robin condition is the only way which provides the possibility to increase the optimality of the Schwarz DDM algorithm.
- Thanks to Robin interface condition, an approximated DtN (Dirichlet-to-Neumann) operator is constructed by purely algebraic means.
- A comparison between several type of transmission interface conditions is done, in both compressible and nearly-incompressible materials, which will show the efficiency of Optimized transmission conditon in Schwarz DD algorithm. The improvement over classical interface conditions is even more significant when approaching the nearly-incompressible limit.

The Maple computations are also presented in the appendix A. This will allow the reader to reproduce all the presented results throughout this chapter.

In chapter 3, the Lions' algorithm is introduced. It is an improvement of the Schwarz algorithm by using the optimized transmission conditon developed in chapter 2. The main objective in this chapter is to give a theory for Lion's algorithm that will be the genuine counterpart of the theory developed over the years for the Schwarz algorithm. Then an emphasis is put into their application to problems with (nearly)-incompressible nature.

- We recall the continuous version of Lions' algorithm as well as the algebraic equivalent version ORAS (Optimized Restricted Additive Schwarz) algorithm.

- A symmetric variant of the ORAS (Optimized Restricted Additive Schwarz) algorithm is proposed, called SORAS.
- We set a mathematical framework for the algorithm. In this part we will see that Fictitious Space Lemma is the appropriate theoretical framework.
- We build an adaptive coarse space which leads to a Two-Level method "SORAS-GenEO2" with prescribed targeted convergence.
- We show that the developed method is equally applicable to saddle point problems.
- Finally in order to show the robustness of the developed Two-Level optimized Schwarz DD method, we perform weak scalability test with elasticity problem on a highly heterogeneous material, that include both compressible and nearly incompressible material. Strong scalability tests are also performed for an incompressible Stokes problem and good scaling results are observed on up to 8192 processors.

In chapter 4, the main goal is to deal efficiently with non linear unstationary fluid dynamics problems when performing large-scale simulations. We will see how we apply the Optimized Schwarz method, developed throughout the previous chapters to Navier-Stokes equations. The chapter is subdivided as follows

- Navier-Stokes equations are presented as well as the context of the Turek flows benchmark [STDKR96].
- Deflation technic is recalled. The unstationary Navier-Stokes equations require to solve multiple linear systems. Finding a way to take advantage from the constructed Krylov subspace becomes a necessity. For this we equip the one level preconditioners ORAS and SORAS with a recycling & deflation technic, which consists in recycling Krylov subspace.
- To show the efficient behavior of the developed optimized Schwarz DD method, we perform and validate a large-scale simulations on Turek benchmark. Optimal scaling results are observed up to 16384 processors.

Publication This thesis led to the following publications

R. Haferssas, P. Jolivet and F Nataf, "A robust coarse space for Optimized Schwarz methods SORAS-GenEO-2", 2015,C. R. Math. Acad. Sci. Paris.

R. Haferssas, P. Jolivet and F Nataf, "An additive Schwarz method type theory for Lions' algorithm and Optimized Schwarz Methods", https://hal.archives-ouvertes.fr/hal-01100926 , 2016,submitted.

Chapter 2

Domain Decomposition Methods

Contents

2.0.1 Original Schwarz methods

The first decomposition method has been introduced by H.A. Schwarz in 1870, see the detailed historical presentation [GW14] by Martin Gander. In that time, the mathematical world was still debating on a particular question: how to rigorously establish the well-posedness of the Laplace's equation with a prescribed boundary conditions, i.e. Laplace's problem written in the following (2.1).

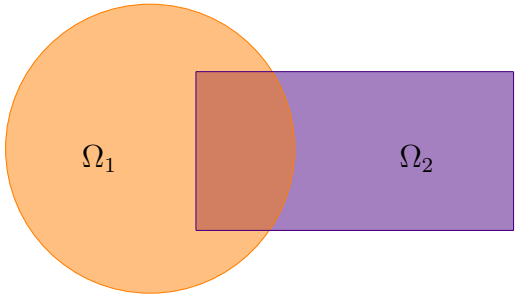


Fig 2.1 – Original Domain Decomposition (by courtesy of P. Jolivet).

Given a boundary condition **g**, find $u : \Omega \longrightarrow \mathbb{R}$ such that

$$\begin{cases} -\Delta u = 0 \text{ in } \Omega \\ \quad u = g \text{ on } \partial\Omega \end{cases} \tag{2.1}$$

Fourier (1807) gave a proof with solution formula for rectangular domain using Fourier series and Poisson (1815) for circular domain, but the question remained open for more complex geometry as shown in figure. Few years later, Schwarz proposed his original idea which consists in subdividing the domain Ω into two regular domains Ω_1 and Ω_2 (rectangle and disk) in order to be able to use some available tools and give a constructive proof to the Laplace's problem.

Definition 1 (Alternating Schwarz algorithm)
The domain Ω is subdivided as $\Omega = \Omega_1 \cup \Omega_2$. Schwarz idea consists in solving alternatively Laplace's problem in each subdomain.

Given (u_1^n, u_2^n), *find* (u_1^{n+1}, u_2^{n+1}) *such that*

$$\begin{cases} -\Delta u_1^{n+1} = 0 & \text{in } \Omega_1 \\ u_1^{n+1} = g & \text{on } \partial\Omega_1 \cap \partial\Omega \\ u_1^{n+1} = u_2^n & \text{on } \partial\Omega_1 \setminus \partial\Omega \end{cases} \qquad \begin{cases} -\Delta u_2^{n+1} = 0 & \text{in } \Omega_2 \\ u_2^{n+1} = g & \text{on } \partial\Omega_2 \cap \partial\Omega \\ u_2^{n+1} = u_1^{n+1} & \text{on } \partial\Omega_2 \setminus \partial\Omega \end{cases}$$

Dirichlet conditions are used as transmission conditions.

When geometry is very simple, using Fourier transform in each subdomain, we can prove that the algorithm initialized with u_2^0 converges toward the solution of (2.1) and thus the well-posedness of the Laplace's problem follows.

Probably without knowing it, Hermann Amandus Schwarz established with this method the pioneer work in the area of Domain Decomposition Methods. Thus, with the emergence of parallel architectures, his algorithm acquired a particular interest. Since then, many works have been addressed on it, one of the first work concerning the sequential aspect, since at each iteration the solution of the Laplace's problem in Ω_2 depends on the solution on the subdomain Ω_1 by the transmission conditions at the same iteration.
A small modification was given by P L. Lions in [Lio89] where the classical sequential algorithm approach evolved to the following fully parallel one.

Definition 2 (Parallel Schwarz algorithm)
Ones the domain Ω is subdivided as $\Omega = \Omega_1 \cup \Omega_2$. Schwarz algorithm bis reads
Given (u_1^n, u_2^n), *find* (u_1^{n+1}, u_2^{n+1}) *such that*

$$\begin{cases} -\Delta u_1^{n+1} = 0 & \text{in } \Omega_1 \\ u_1^{n+1} = g & \text{on } \partial\Omega_1 \cap \partial\Omega \\ u_1^{n+1} = u_2^n & \text{on } \partial\Omega_1 \setminus \partial\Omega \end{cases} \qquad \begin{cases} -\Delta u_2^{n+1} = 0 & \text{in } \Omega_2 \\ u_2^{n+1} = g & \text{on } \partial\Omega_2 \cap \partial\Omega \\ u_2^{n+1} = u_1^n & \text{on } \partial\Omega_2 \setminus \partial\Omega \end{cases}$$

This algorithm can be seen as a generalization of Jacobi method. Formally, it is easy to prove that if convergence occurs, local solutions match in the overlap $\Omega_1 \cap \Omega_2$. Indeed we know that

in the intersection (overlap) $\Omega_{1,2} = \Omega_1 \cup \Omega_2$ the solution u^∞ is the same, i.e. $u^\infty = u_1^\infty = u_2^\infty$. Let $e^\infty := u_1^\infty - u_2^\infty$. Since $e^\infty = 0$ in $\partial\Omega_{1,2}$ and, on the other hand, it is known by linearity of the Poisson operator that $\Delta e^\infty = 0$ in $\Omega_{1,2}$, then e^∞ solves the following boundary value problem

$$\begin{cases} -\Delta e^\infty = 0 \text{ in } \Omega_{1,2} \\ \quad e^\infty = 0 \text{ on } \partial\Omega_{1,2} \end{cases} \tag{2.2}$$

which has zero as unique solution.

Remark 1
The original alternative Schwarz algorithm was made for the solution of a boundary value problems for harmonic functions. The typical problem is the Laplace's equation with prescribed boundary condition, although in the literature, the Schwarz algorithm is often presented in the case of the Poisson's problem.

Convergence factor The convergence of the Schwarz algorithm can be established by several methods, namely by using the maximum principle, Hilbert spaces as pointed out in [Lio89], or using basic Fourier tools. In order to have an idea about the convergence of the method, we recall here from [DJN15] a Fourier convergence analysis, which was conducted on the following elliptic problem

$$\begin{cases} \eta u + \Delta u = f \quad \text{in } \mathbb{R}^2, \\ u \text{ is bounded at infnity.} \end{cases} \tag{2.3}$$

The domain $\Omega = \mathbb{R}^2$ is decomposed into two subdomains $\Omega_1 =]-\infty, \delta] \times \mathbb{R}$ and $\Omega_2 = [0, +\infty[\times\mathbb{R}$ with $\delta \geqslant 0$ the overlapping parameter. Fourier analysis provides the convergence factor ρ, where

$$\rho(k, \delta) := e^{-\lambda(k)\delta}, \quad \text{with } \lambda(k) = \sqrt{\eta + k^2}. \tag{2.4}$$

The unique key of the convergence is the overlap δ, since the convergence becomes faster with increasing the size of the overlap δ.

2.0.2 Generalization of Schwarz algorithm

Let $\mathcal{L}$ a linear operator, we consider a decomposition of the domain $\Omega = \Omega_1 \cup \Omega_2$ (see figure 2.1). In order to solve the following general linear problem

$$\begin{cases} \mathcal{L}u = f \text{ in } \Omega \\ \quad u = 0 \text{ on } \partial\Omega\,. \end{cases} \tag{2.5}$$

Schwarz algorithm applied to problem (2.5) reads:

Given (u_1^n, u_2^n), find (u_1^{n+1}, u_2^{n+1}) such that for $i, j = 1, 2$

$$\begin{cases} \mathcal{L}(u_i^{n+1}) = f & \text{in } \Omega_i \\ \quad u_i^{n+1} = 0 & \text{on } \partial\Omega_i \cap \partial\Omega \\ \quad u_i^{n+1} = u_j^n & \text{on } \partial\Omega_i \cap \overline{\Omega}_j\,. \end{cases} \tag{2.6}$$

We can write an algorithm which is related to classical Schwarz algorithm (2.6) but instead of acting on local functions u_i^n, it acts on the global function u^n. Through this section, we will

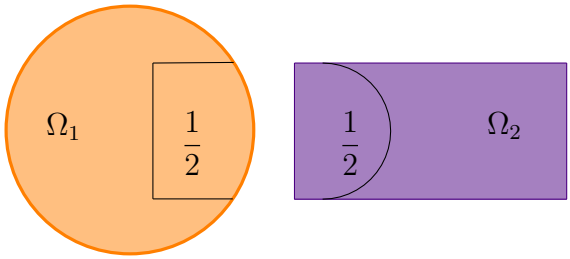


Fig 2.2 – Partition of unity

introduce two versions.
Before going to the writing of global algorithms, we need to define extension, restriction and partition of unity operators.

Definition 3 (Extension & Restriction operators)
Let E_i be an extension operator defined as $E_i(\omega_i) : \Omega \longrightarrow \mathbb{R}$ such that E_i is the extension function of $\omega_i : \Omega_i \longrightarrow \mathbb{R}$ by zero outside Ω_i.

Definition 4 (Partition of unity)
We define the partition of unity $\chi_i : \Omega_i \longrightarrow \mathbb{R}$, such that $\chi \geqslant 0$, $\chi_i(\mathrm{x}) = 0$ on $\partial\Omega_i \backslash \partial\Omega$ and

$$\omega = \sum_{i=1}^{2} E_i(\chi_i \omega_{|\Omega_i}),$$

where $\omega : \Omega \longrightarrow \mathbb{R}$.

2.0.3 From Classical Schwarz algorithm to a global Schwarz algorithm

Let $\mathcal{L}$ a linear operator, we consider a decomposition of the domain $\Omega := \Omega_1 \cup \Omega_2$.
Given u^n an approximation of the solution to the problem (2.5), find u^{n+1} by first solving in parallel

$$\begin{cases} \mathcal{L}(\mathrm{v}_i^{n+1}) = \mathrm{f} & \text{in } \Omega_i \\ \mathrm{v}_i^{n+1} = 0 & \text{on } \partial\Omega_i \cap \partial\Omega \\ \mathrm{v}_i^{n+1} = \mathrm{u}^n & \text{on } \partial\Omega_i \cap \overline{\Omega}_j \end{cases} \tag{2.7}$$

and then glue the local solution as

$$\mathrm{u}^{n+1} = \sum_{i=1}^{2} \mathrm{E}_i(\chi_i \mathrm{v}_i^{n+1}). \tag{2.8}$$

At this point, we can recall the following results

Lemma 5
The global algorithm defined in definition (2.7) and (2.8) which acts on the global function u^n is equivalent to the parallel Schwarz algorithm defined in (2.6) which iterates on u_i^n, that is to say:

$$\mathrm{u}^n = \sum_{i=1}^{2} \mathrm{E}_i(\chi_i \mathrm{u}_i^n)$$

holds at every step n of the algorithm, provided it holds at the initial step $n = 0$.

Proof. The proof can be conducted by induction. First by hypothesis the property is true for

$$\mathrm{u}^0 = \sum_{i=1}^{2} \mathrm{E}_i(\chi_i \mathrm{u}_i^0),$$

and holds for the nth term. Then using the fact that $\chi_1 \equiv 0$ and $\chi_2 \equiv 1$ on $\partial\Omega_1 \cap \overline{\Omega}_2$ thus by definition v_1^{n+1} is solution to the problem (2.6)

$$\begin{cases} \mathcal{L}(\mathrm{v}_1^{n+1}) = \mathrm{f} & \text{in } \Omega_1 \\ \mathrm{v}_1^{n+1} = 0 & \text{on } \partial\Omega_1 \cap \partial\Omega \\ \mathrm{v}_1^{n+1} = \mathrm{u}^n = \displaystyle\sum_{i=1}^{2} \mathrm{E}_i(\chi_i \mathrm{u}_i^n) = \mathrm{u}_2^n & \text{on } \partial\Omega_1 \cap \overline{\Omega}_2\,, \end{cases}$$

hence $\mathrm{v}_1^{n+1} = \mathrm{u}_1^{n+1}$, the same proof holds for $\mathrm{v}_2^{n+1} = \mathrm{u}_2^{n+1}$ in Ω_2, and we can say that

$$\mathrm{u}^{n+1} = \sum_{i=1}^{2} \mathrm{E}_i(\chi_i \mathrm{v}_i^{n+1}) = \sum_{i=1}^{2} \mathrm{E}_i(\chi_i \mathrm{u}_i^{n+1})\,.$$

This completes the proof. ■

With the view to move to the total algebraic formulation of Schwarz algorithm (2.6), we express algorithm (2.7) and (2.8) with residuals $r^n = \mathrm{f} - \mathcal{L}(\mathrm{u}^n)$, then we will be able with Algorithm 1 to introduce a continuous version of the so-called called Restrictive Additive Schwarz (RAS) algorithm [CS99].

2.0.4 From Classical Schwarz algorithm to global RAS algorithm

In this section, we introduce the RAS (Restrictive Additive Schwarz) algorithm in order to recall the existent relation between Schwarz algorithm and RAS.

Lemma 6
Continuous algorithm given by Algorithm 1 called Restrictive Additive Schwarz is equivalent to Schwarz (RAS) algorithm defined in (2.6).

Proof. The main points to prove is the following equality

$$\mathrm{u}^n = \mathrm{E}_1(\chi_1 \mathrm{u}_1^n) + \mathrm{E}_2(\chi_2 \mathrm{u}_2^n),$$

where $\mathrm{u}_1^n, \mathrm{u}_2^n$ are given by Lions' algorithm in (2.6) and u^n is a solution given by RAS Algorithm 1. To do this, we conduct a proof by induction. First, from the initial guess, which satisfies

$$\mathrm{u}^0 = \mathrm{E}_1(\chi_1 \mathrm{u}_1^0) + \mathrm{E}_2(\chi_2 \mathrm{u}_2^0),$$

we assume that the property is true till the nth iteration of the algorithms

$$\mathrm{u}^n = \mathrm{E}_1(\chi_1 \mathrm{u}_1^n) + \mathrm{E}_2(\chi_2 \mathrm{u}_2^n),$$

Algorithm 1 continuous RAS algorithm

1: Compute residual
$$r^n = \mathrm{f} - \mathcal{L}(\mathrm{u}^n)$$
2: **for** each subdomain $i = 1, 2$, to compute a local correction **do**
$$\begin{cases} \mathcal{L}(\mathrm{v}_i^{n+1}) = r^n & \text{in } \Omega_i \\ \mathrm{v}_i^{n+1} = 0 & \text{on } \partial\Omega_i \cap \partial\Omega \\ \mathrm{v}_i^{n+1} = 0 & \text{on } \Gamma_{ij} = \partial\Omega_i \cap \overline{\Omega}_j \end{cases}$$
3: **end for**
4: Update u^n by
$$\mathrm{u}^{n+1} = \mathrm{u}^n + \mathrm{E}_1(\chi_1 \mathrm{v}_1^{n+1}) + \mathrm{E}_2(\chi_2 \mathrm{v}_2^{n+1})$$
where E_i and χ_i are respectively the extension operators and the partition of unity defined respectively in Definitions 3 and 4.

from the algorithm (2.6) we have

$$\mathrm{u}^{n+1} = \mathrm{E}_1(\chi_1(\mathrm{u}^n_{|\Omega_1} + \mathrm{v}_1^n)) + \mathrm{E}_2(\chi_2(\mathrm{u}^n_{|\Omega_2} + \mathrm{v}_2^n)).$$

Now we show that
$$\mathrm{u}_1^{n+1} = \mathrm{u}^n_{|\Omega_1} + \mathrm{v}_1^n \,.$$

This requires to prove that $\mathrm{u}^n_{|\Omega_1} + \mathrm{v}_1^n$ is a solution to the problem (2.6), we have

$$\begin{cases} \mathcal{L}(\mathrm{u}^n_{|\Omega_1} + \mathrm{v}_1^n) = \mathcal{L}(\mathrm{u}^n_{|\Omega_1}) + r^n = \mathcal{L}(\mathrm{u}^n_{|\Omega_1}) + f - \mathcal{L}(\mathrm{u}^n_{|\Omega_1}) = \mathrm{f} & \text{in } \Omega_1 \\ \mathrm{u}^n_{|\Omega_1} + \mathrm{v}_1^n = 0 & \text{on } \partial\Omega_1 \cap \partial\Omega \\ \mathrm{u}^n_{|\Omega_1} + \mathrm{v}_1^n = \mathrm{u}^n_{|\Omega_1} & \text{on } \Gamma_{12} \,. \end{cases}$$

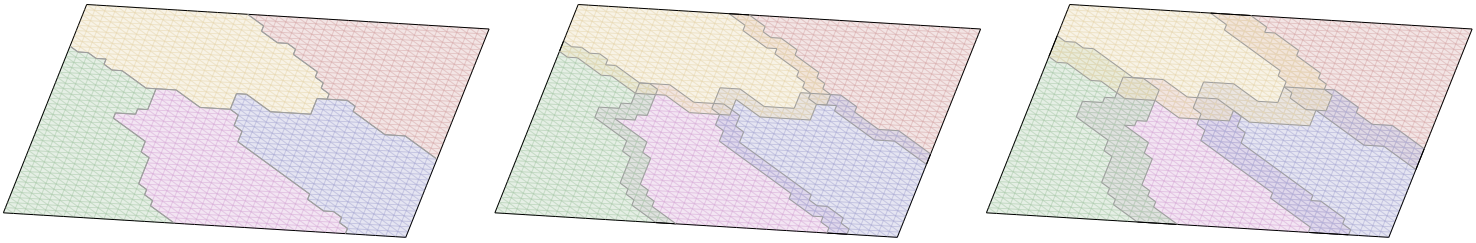
The last thing to prove is
$$\mathrm{u}^n_{|\Omega_1} = \mathrm{u}_2^n \,.$$
from the induction hypothesis $\mathrm{u}^n = \mathrm{E}_1(\chi_1 \mathrm{u}_1^n) + \mathrm{E}_2(\chi_2 \mathrm{u}_2^n)$. In addition to that, on the interface Γ_{12} we have $\chi_1 \equiv 0$ thus $\chi_2 \equiv 1$ then

$$\mathrm{u}^n_{|\Omega_1} = \sum_{i=1}^{2} \mathrm{E}_i(\chi_i \mathrm{u}_i^n) = \mathrm{u}_2^n \quad \text{on } \Gamma_{12} \,.$$

Hence, we can say that $\mathrm{u}^n_{|\Omega_1} + \mathrm{v}_1^n = \mathrm{u}_1^{n+1}$ which satisfies (2.6). The same proof holds for every $\{\Omega_i\}_{1 \leqslant i \leqslant 2}$, where $\mathrm{u}^n_{|\Omega_i} + \mathrm{v}_i^n = \mathrm{u}_i^{n+1}$ which finally allows us to write

$$\mathrm{u}^{n+1} = \sum_{i=1}^{2} \mathrm{E}_i(\chi_i \mathrm{u}_i^{n+1}).$$

This ends the proof of equivalence between classical Schwarz algorithm and continuous RAS algorithm. ■



(a) $l = 0$ **(b)** $l = 1$ **(c)** $l = 2$

Fig 2.3 – Partition of $\Omega = [0;1]^2$ into $N = 5$ subdomains with different values for the overlap parameter(by courtesy of P. Jolivet).

2.0.5 Going to multiple arbitrary decomposition with algebraic formulation

Throughout this section, we will introduce more general Schwarz algorithm for multiple subdomains with a purely algebraic formulation, for this reason, we need to translate some ingredients defined before such as the extension and partition of unity operator to an algebraic form

- Let Ω be an arbitrary domain discretized using a mesh $\mathcal{T}_h$, Ω is decomposed into N subdomains $(\Omega_i)_{1\leq i\leq N}$ so that all subdomains are a union of cells of the mesh $\mathcal{T}_h$.

$$\Omega := \bigcup_{i=1}^{N} \{\overline{\Omega}_i\} \quad \text{where } \Omega_i \cap \Omega_j = \emptyset, \quad i \neq j.$$

- This decomposition induces a natural decomposition of the set of indices $\mathcal{N}$ into N subsets of indices $(\mathcal{N}_i)_{1\leq i\leq N}$:

$$\mathcal{N} := \bigcup_{i=1}^{m} \mathcal{N}_i \quad .$$

- In the case of overlapping decomposition, we extend each subdomain Ω_i to a domain Ω_i^δ by adding one or several layers (which can be adjacent fine grid, see figure 2.3), this will induce the extension of each subset $\mathcal{N}_i$ with its direct neighbors, which will form $\mathcal{N}_i^\delta$, δ is the size of overlap.

- We consider a vector $\mathbf{U} \in \mathbb{R}^{\#\mathcal{N}}$, then for all $1 \leq i \leq N$, let R_i be matrix from $\mathbb{R}^{\#\mathcal{N}}$ to the subset $\mathbb{R}^{\#\mathcal{N}_i^\delta}$ the algebraic counterpart part of restriction operator defined in definition 3

- Let D_i be a diagonal matrix of size $\#\mathcal{N}_i^\delta \times \#\mathcal{N}_i^\delta$, the algebraic counterpart part of partition of unity operator defined in definition 4. see figure 2.2. So, that we have a partition of unity at the algebraic level,

$$\sum_{i=1}^{N} R_i^T D_i R_i = I_d \,,$$

  where $I_d \in \mathbb{R}^{\#\mathcal{N}\times\#\mathcal{N}}$ is the identity matrix.

- For all subdomains $1 \leq i \leq N$, let A_i be a SPD matrix of size $\#\mathcal{N}_i^\delta \times \#\mathcal{N}_i^\delta$, which comes typically from the discretization of local boundary value problems.

One-Level Schwarz Domain Decomposition as preconditioner

Now, we can introduce the algebraic counterpart of the Additive Schwarz method (ASM) and Restricted Additive Schwarz (RAS) see [CFS98; CS99] algorithm as preconditioners for the following fixed-point problem

$$\mathbf{U}^{n+1} = \mathbf{U}^n + M^{-1}(\mathbf{F} - A\mathbf{U}^n)\,,$$

where M^{-1} reads

- For **ASM**
$$M^{-1}_{ASM,1} := \sum_{i=1}^{N} R_i^T A_i^{-1} R_i\,. \quad \text{with } A_i = R_i A R_i^T\,.$$
- For **RAS**
$$M^{-1}_{RAS,1} := \sum_{i=1}^{N} R_i^T D_i A_i^{-1} R_i\,. \quad \text{with } A_i = R_i A R_i^T\,.$$

Two-Level Schwarz Domain Decomposition as preconditioner

As it is well known, the drawback of a One-Level methods is that its convergence rate depends on the number of subdomains. This dependance is harmful to some PDEs problems such as the one provided by solid or fluid mechanics, where particularly the One-Level methods scales poorly for large problems. This can be explained by a lack of global communication between subdomains since at each iteration subdomains have exchanges with its direct neighbors only.

One of the proposed remedies consists in adding an additional space, which will have the task to spread the informations to all subdomains at each iteration. Thus, the enrichment with a coarse space is now a classical way of diminishing this lack of robustness. Such methods are called two level methods. One of the first work which introduced this kind of remedy was done by Nicolaides [Nic87]. This kind of remedy is closely related to deflation technics in linear algebra used in [TNVE09]. It is connected as well to augmented or recycled Krylov space methods see e.g. [EG00; PSMJM06a; Saa97]. We can find similar approach in multigrid community such as the one proposed in [Vas08].

Having set the coarse space $\mathcal{H}_0$. Let $Z \in \mathbb{R}^{\#\mathcal{N}\times\#\mathcal{N}_0}$, be a rectangular matrix whose columns correspond to a set of vectors which span the coarse space.

In the following, we introduce the several existing algebraic ways to enrich and accelerate the one-level preconditioner M^{-1} (e.g. $M^{-1} = M^{-1}_{ASM,1}$) by the coarse operator.

Definition 7 (Two-Level preconditioner)
Two level domain decomposition method can be defined as

- $M^{-1}_{2,AD} := Q + M^{-1}$.
- $M^{-1}_{2,A-DEF_1} := M^{-1}(I - AQ) + Q$.
- $M^{-1}_{2,A-DEF_2} := (I - QA)M^{-1} + Q$.
- $M^{-1}_{2,BNN} := (I - QA)M^{-1}(I - AQ) + Q$.

Where $Q = Z(Z^T AZ)^{-1} Z^T$ *is the coarse operator and* M^{-1} *is the one level preconditioner. We recall that in the case of Additive Schwarz method* $M^{-1}_{ASM,1} := \sum_{i=1}^{N} R_i^T A_i^{-1} R_i$. *with* $A_i = R_i A R_i^T$.

Remark 2
We can make some straightforward observations

- The coarse operator Q couples all the subdomains but the square matrix $Z^T AZ$ has a much smaller dimension than that of A. The extra cost due to its inversion is negligible comparing to the given scalability.
- The structure of the two-level preconditioner M_2^{-1} is the same as the one of one-level preconditioner M_1^{-1}.

Chapter 3

Optimized Domain Decomposition Methods

Contents

3.1 Introduction

The interest for domain decomposition methods is constantly increasing and Schwarz method is particularly attracting much attention from researcher in parallel computations. The advantage in parallelism, wide applicability to PDEs and great flexibility in implementation make the Schwarz method very competitive. During the last decades various extensions based upon an optimal convergence consideration were proposed. It is noticed that most of significant improvements in the convergence rate of Schwarz method depend intrinsically on the considered type of transmission conditions between subdomains. P.L. Lions [Lio90] in 1990 proposed to use Robin transmission condition instead of Dirichlet condition in non-overlapping Schwarz method to obtain a convergent algorithm. This led to an other possibility to improve the optimality of Schwarz method, In [GHN01] M.J. Gander et al. introduced for the first time Optimized Schwarz Method using Robin transmission condition.

Pursuing in that direction, throughout this chapter we give some details on the well-known

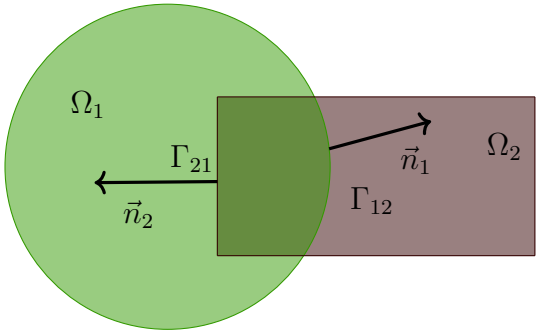


Fig 3.1 – 2D domain decomposition for Lions algorithm

P.L. Lions algorithm from its origin, then we investigate several type of transmission conditions for Schwarz method focusing on Elasticity problem and particularly on the incompressible limit for nearly incompressible materials. An extension of P.L. Lions algorithm is done for systems of equations with an improvement of the optimality via new coefficients.

3.2 Robin interface conditions & Lions' algorithm

Robin transmission condition for Schwarz algorithm has been introduced and studied for non-overlapping domain decomposition in the case of Poisson's problem in [Lio90], however, they can also be applied to overlapping subdomain see figure 3.1, where the form of Robin transmission condition reads
Let α be a positive parameter

$$\begin{aligned}\left(\frac{\partial}{\partial n_1}+\alpha\right)\mathrm{u}_1^{n+1} &= \left(\frac{\partial}{\partial n_1}+\alpha\right)\mathrm{u}_2^{n} \quad \text{on } \partial\Omega_1\cap\bar{\Omega}_2\,,\\ \left(\frac{\partial}{\partial n_2}+\alpha\right)\mathrm{u}_2^{n+1} &= \left(\frac{\partial}{\partial n_2}+\alpha\right)\mathrm{u}_1^{n} \quad \text{on } \partial\Omega_2\cap\bar{\Omega}_1\,,\end{aligned} \tag{3.1}$$

where n_1 and n_2 are the outward normals on the boundary of each subdomain.

In the the following, we give the P.L. Lions algorithm as it was introduced in [Lio90] for non-overlapping subdomain, then recall its application in case of overlapping subdomains.

Definition 8 (P.L. Lions' Algorithm)
Let Ω be a domain decomposed in m non overlapping subdomains Ω_i $1 \leqslant i \leqslant m$. Then P.L.Lions' algorithm reads

Given u_i^n, *find* u_i^{n+1} *such that*

$$\begin{cases} -\Delta \mathrm{u}_i^{n+1} = \mathrm{f} & \text{in } \Omega_i \\ \mathrm{u}_i^{n+1} = 0 & \text{on } \partial\Omega_i \cap \partial\Omega \\ \left(\dfrac{\partial}{\partial n_{ij}} + \alpha_{ij}\right) \mathrm{u}_i^{n+1} = \left(\dfrac{\partial}{\partial n_{ij}} + \alpha_{ij}\right) \mathrm{u}_j^n \quad \text{on } \Gamma_{ij} \quad \forall j, \quad 1 \leqslant j \leqslant m \quad j \neq i\,, \end{cases} \tag{3.2}$$

where n_{ij} *are the outward normals on the boundary* $\Gamma_{ij} := \Omega_i \cap \Omega_j$ *of each subdomain* Ω_i *and* $\alpha_{ij} = \alpha_{ji}$ *for all* $\quad 1 \leqslant i \neq j \leqslant m$.

In the same paper P.L. Lions [Lio90] gives a first convergence proof for the case of a non-overlapping decomposition. We recall this important result in the following theorem

Theorem 9
For all $\quad 1 \leqslant i \leqslant m$, u_i^n *converges in* $H^1(\Omega_i)$ *(i.e.* $\lim\limits_{n\to\infty} \|\mathrm{u}_i^n - \mathrm{u}_{|\Omega_i}\|_{H^1(\Omega_i)} = 0$*).*
In particular for all $j \neq i$, $\mathrm{u}_{i|\gamma_{ij}}^n$ *converge to* $\mathrm{u}_{i|\gamma_{ij}}$ *in* $H^{\frac{1}{2}}$.

In the same work [Lio90], the algorithm was extended to the following more general second-order elliptic operator

$$\mathcal{L}(\mathrm{u}) := -\Delta \mathrm{u} + b(\mathrm{x}) \cdot \nabla \mathrm{u} + c(\mathrm{x})\,,$$

where the Schwarz algorithm reads for $1 \leqslant i \leqslant m$. Given u_i^n, find u_i^{n+1} such that

$$\begin{cases} -\Delta \mathrm{u}_i^{n+1} + b(\mathrm{x}) \cdot \nabla \mathrm{u}_i^{n+1} + c(\mathrm{x}) = \mathrm{f} & \text{in } \Omega_i \\ \mathrm{u}_i^{n+1} = 0 & \text{on } \partial\Omega_i \cap \partial\Omega \\ \left(\dfrac{\partial}{\partial n_{ij}} + \alpha_{ij}\right) \mathrm{u}_i^{n+1} = \left(\dfrac{\partial}{\partial n_{ij}} + \alpha_{ij}\right) \mathrm{u}_j^n \quad \text{on } \Gamma_{ij} \quad \forall j, \quad 1 \leqslant j \leqslant m \quad j \neq i\,. \end{cases}$$

It is also proven that u_i^n converges to $\mathrm{u}_{|\partial\Omega_i}$ in $L^2(\partial\Omega_i)$ and u_i^n converges to $\mathrm{u}_{|\Omega_i}$ in $L^2(\Omega_i)$.

The algorithm and convergence proof were extended to Helmholtz equation by B. Després in [Des93] and also for time-harmonic Maxwell equations in [DJR92].

Remark 3
Since the first work of P-L. Lions, the effective choice of α_{ij} has always been an open question for many PDEs operators.

Remark 4
The main feature of this results is that it is very general, since it considers an arbitrary number of subdomains. However, besides the fact that it concerns only non-overlapping decomposition, it does not give any idea on the estimate of the rate of convergence for the discrete algorithm.

3.2.1 Convergence factor

In order to estimate the rate of convergence when using Robin transmission condition, as shown in [DJN15], other convergence analysis were done for both overlapping and non-overlapping decompositions in simple cases for the following Lions' algorithm

Given $(\mathrm{u}_1^n, \mathrm{u}_2^n)$, find $(\mathrm{u}_1^{n+1}, \mathrm{u}_2^{n+1})$ such that

$$\begin{cases} \eta \mathrm{u}_1^{n+1} - \Delta \mathrm{u}_1^{n+1} = \mathrm{f} & \text{in } \Omega_1 \\ \mathrm{u}_1^{n+1} = 0 & \text{on } \partial\Omega_1 \cap \partial\Omega \\ \left(\dfrac{\partial}{\partial n_1} + \alpha\right) \mathrm{u}_1^{n+1} = \left(\dfrac{\partial}{\partial n_1} + \alpha\right) \mathrm{u}_2^{n} & \text{on } \Omega \setminus \Omega_1 \end{cases}, \tag{3.3}$$

and

$$\begin{cases} \eta \mathrm{u}_2^{n+1} - \Delta \mathrm{u}_2^{n+1} = \mathrm{f} & \text{in } \Omega_2 \\ \mathrm{u}_2^{n+1} = 0 & \text{on } \partial\Omega_2 \cap \partial\Omega \\ \left(\dfrac{\partial}{\partial n_2} + \alpha\right) \mathrm{u}_2^{n+1} = \left(\dfrac{\partial}{\partial n_2} + \alpha\right) \mathrm{u}_1^{n} & \text{on } \Omega \setminus \Omega_2. \end{cases} \tag{3.4}$$

Where α is a positive parameter, n_1 and n_2 are the outward normals on the boundary of each subdomain.

Using Fourier analysis in the case where the $\Omega = \mathbb{R}^2$ is decomposed into two half planes with an overlap of size $2\delta \geq 0$ (see figure 3.2), the convergence factor ρ in the Fourier space (dual variable is denoted k) has the following form

$$\rho(k, \delta; \alpha) := \left| \frac{\lambda(k) - \alpha}{\lambda(k) + \alpha} \right| \mathrm{e}^{-2\lambda(k)\delta} \quad \text{with} \quad \lambda(k) = \sqrt{\eta + k^2} \quad \text{with} \quad \alpha > 0. \tag{3.5}$$

Remark 5
We can distinguish three features from the above result

- We can already see that even if there is no overlap i.e. $\delta = 0$, $|\rho(k, 0; \alpha)| < 1$, the method is still convergent.
- The choice of the parameter α can be done based on this formula, see [DJN15] and references therein.
- In the particular case of $\rho(k, \delta; \infty)$, the convergence factor corresponds to the use of Dirichlet condition and in the case of $\rho(k, \delta; 0)$ it will corresponds to the use of Neumann condition, we can already notice that Robin condition is more general.

3.3 Elasticity equations

Throughout this section, we investigate several transmission condition in Schwarz method for elasticity problem for both compressible and almost incompressible materials.

We can carry out a similar reasoning using Fourier analysis on the elasticity equations and study the convergence behavior of Schwarz domain decomposition methods with several types of transmission conditions, first with Dirichlet conditions, then Neumann conditions and more generally with Robin conditions. This shows us how the frequency components of the convergence error are damped with respect to the applied interface conditions.

To begin, we first write the elasticity systems in the framework of Schwarz method, then perform a Fourier analysis in order to have an idea on the convergence factor.

Consider the following elasticity equations

$$\begin{cases} \mathcal{L}(\underline{\mathrm{u}}) := \underline{\nabla} \cdot \underline{\underline{\sigma}}_S(\underline{\mathrm{u}}(\mathrm{x})) & = \underline{\mathrm{f}} \text{ in } \Omega \\ \underline{\mathrm{u}}(\mathrm{x}) \text{ bounded at infinity,} \end{cases} \tag{3.6}$$

with

$$\underline{\underline{\sigma}}_S = \lambda \,\mathrm{tr}(\underline{\underline{\varepsilon}})\mathrm{Id} + 2\mu\underline{\underline{\varepsilon}} \quad \text{and} \quad \underline{\underline{\varepsilon}}(\underline{\mathrm{u}}) = \frac{1}{2}\left(\underline{\underline{\nabla}}\,\underline{\mathrm{u}} + (\underline{\underline{\nabla}}\,\underline{\mathrm{u}})^T\right),$$

λ and μ are the Lamé parameters, defined with Young's modulus E and Poisson ratio ν

$$\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)} \quad \text{and} \quad \mu = \frac{E}{2(1+\nu)}.$$

For the sake of simplicity, we place the equations in a two dimensional framework where $\Omega = \mathbb{R}^2$. Then from equation (3.6), we get for a 2D displacement vector $\underline{\mathrm{u}} = (u, v)^\intercal$ and right hand side $\underline{\mathrm{f}} = (f_1, f_2)^\intercal$ the following

$$\underline{\underline{\sigma}}_S(\underline{\mathrm{u}}) = \begin{bmatrix} \lambda\left(\partial_x u + \partial_y v\right) + 2\mu\partial_x u & \mu\left(\partial_y u + \partial_x v\right) \\ \\ \mu\left(\partial_y u + \partial_x v\right) & \lambda\left(\partial_x u + \partial_y v\right) + 2\mu\partial_y v \end{bmatrix}$$

and

$$\underline{\nabla} \cdot \underline{\underline{\sigma}}_S(\underline{\mathrm{u}}) = \begin{bmatrix} (\lambda + 2\mu)\,\partial_{xx} u + (\lambda + \mu)\,\partial_{xy} v + \mu\partial_{yy} u \\ \\ (\lambda + 2\mu)\,\partial_{yy} v + (\lambda + \mu)\,\partial_{xy} u + \mu\partial_{xx} v \end{bmatrix}$$

Hence the system of elasticity equation with Saint-Venant Kirchhoff constitutive law reads

$$\mathcal{L}(\underline{\mathrm{u}}) = \begin{cases} -\mu\Delta u - (\lambda+\mu)\partial_x \underline{\nabla} \cdot (\underline{\mathrm{u}}) = f_1 \\ -\mu\Delta v - (\lambda+\mu)\partial_y \underline{\nabla} \cdot (\underline{\mathrm{u}}) = f_2 \end{cases}, \text{ with } \underline{\mathrm{u}} = (u, v)^\intercal. \tag{3.7}$$

In the case where the domain Ω is partitioned into two half planes $\Omega_1 =]-\infty, \delta] \times \mathbb{R}$ and $\Omega_2 = [0, +\infty[\times\mathbb{R}$ with $\delta > 0$ the overlapping parameter, we denote by $\underline{\mathrm{u}}_1 = (u_1, v_1)^\intercal$ and $\underline{\mathrm{u}}_2 = (u_2, v_2)^\intercal$, the solutions to find in each subdomain. $\mathcal{T}$ is an operator to be chosen later, it determines the interface conditions type in the Schwarz algorithm (for example $\mathcal{T} = Id$ gives us the classical Schwarz algorithm). Then consider the following parallel Schwarz algorithm. Given $\underline{\mathrm{u}}_i^n$, find $\underline{\mathrm{u}}_i^{n+1}$ such that

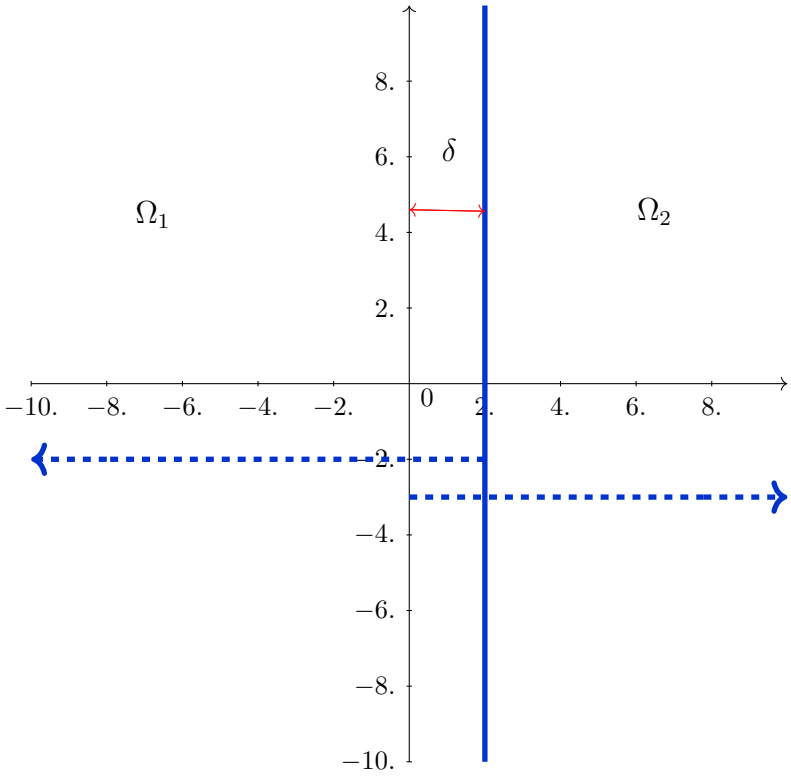


Fig 3.2 – 2D simplified domain decomposition with two half plane, overlap = δ.

$$\begin{cases} \mathcal{L}(\underline{\mathrm{u}}_1^{n+1}) = \mathrm{f}_1 \text{ in } \Omega_1, \\ \underline{\mathrm{u}}_1^{n+1} \text{ bounded at infinity}, \\ \mathcal{T}\underline{\mathrm{u}}_1^{n+1}(\delta, y) = \mathcal{T}\underline{\mathrm{u}}_2^{n}(\delta, y) \quad y \in \mathbb{R}. \end{cases} \qquad \begin{cases} \mathcal{L}(\underline{\mathrm{u}}_2^{n+1}) = \mathrm{f}_2 \text{ in } \Omega_2, \\ \underline{\mathrm{u}}_2^{n+1} \text{ bounded at infinity}, \\ \mathcal{T}\underline{\mathrm{u}}_2^{n+1}(0, y) = \mathcal{T}\underline{\mathrm{u}}_1^{n}(0, y), \quad y \in \mathbb{R}. \end{cases}$$

3.3.1 Fourier framwork

Throughout this section, we study and compare the convergence factor for various existing interface conditions, then we establish the existence of other interface conditions which can increase the optimality.

Firstly, in order to derive the convergence factor we introduce the error $\underline{\mathcal{E}}_i = (u_{\mathcal{E},i}, v_{\mathcal{E},i})^\intercal$ where $\underline{\mathcal{E}}_i = \underline{\mathrm{u}}_i - \underline{\mathrm{u}}_{\Omega_i}$ for $i = 1, 2$. By linearity, the errors satisfy the following algorithm.

Given $\underline{\mathcal{E}}_i^n$, find $\underline{\mathcal{E}}_i^{n+1}$ such that

$$\begin{cases} \mathcal{L}(\underline{\mathcal{E}}_1^{n+1}) = 0 \text{ in } \Omega_1, \\ \underline{\mathcal{E}}_1^{n+1} \text{ bounded at infinity}, \\ \mathcal{T}\underline{\mathcal{E}}_1^{n+1}(\delta, y) = \mathcal{T}\underline{\mathcal{E}}_2^{n}(\delta, y), \ y \in \mathbb{R}, \end{cases} \tag{3.8}$$

and

$$\begin{cases} \mathcal{L}(\underline{\mathcal{E}}_2^{n+1}) = 0 \text{ in } \Omega_2, \\ \underline{\mathcal{E}}_2^{n+1} \text{ bounded at infinity}, \\ \mathcal{T}\underline{\mathcal{E}}_2^{n+1}(0, y) = \mathcal{T}\underline{\mathcal{E}}_1^{n}(0, y), \ y \in \mathbb{R}. \end{cases} \tag{3.9}$$

The convergence of Schwarz algorithms can be analyzed in detail using Fourier transform. In order to use these tools, we shall recall the Fourier transform in our framework. For this, we denote the partial Fourier transform for $\mathrm{f}(\mathrm{x},\mathrm{y}) : \mathbb{R}^2 \longrightarrow \mathbb{R}$ in the y direction by

$$\hat{\mathrm{f}}(\mathrm{x},k) = \mathcal{F}_{\mathrm{y}}(\mathrm{f})(\mathrm{x},k) := \int_{-\infty}^{+\infty} \exp(-ik\mathrm{y})\mathrm{f}(\mathrm{x},\mathrm{y})d\mathrm{y},$$

$i^2 = -1$ and the inverse Fourier transform of $\hat{\mathrm{f}}(\mathrm{x},k)$ by

$$\mathrm{f}(\mathrm{x},\mathrm{y}) = \mathcal{F}^{-1}(\hat{\mathrm{f}})(\mathrm{x},\mathrm{y}) := \frac{1}{2\pi}\int_{-\infty}^{+\infty} \exp(ik\mathrm{y})\hat{\mathrm{f}}(\mathrm{x},k)dk.$$

We perform the Fourier transform of the two equations in (3.7) in the y direction, then, elasticity operator considered in (3.6) can be written for $k \in \mathbb{R}$ as

$$\mathcal{L}(\hat{\underline{\mathcal{E}}}) = \begin{cases} -(2\mu+\lambda)\dfrac{\partial^2 \hat{u}_{\mathcal{E}}}{\partial x^2} + k^2\mu\hat{u}_{\mathcal{E}} - ik(\lambda+\mu)\dfrac{\partial \hat{v}_{\mathcal{E}}}{\partial x} = 0 \\ k^2(2\mu+\lambda)\hat{v}_{\mathcal{E}} - \mu\dfrac{\partial^2 \hat{v}_{\mathcal{E}}}{\partial x^2} - ik(\lambda+\mu)\dfrac{\partial \hat{u}_{\mathcal{E}}}{\partial x} = 0 \end{cases}$$

where $\hat{\underline{\mathcal{E}}} = (\hat{u}_{\mathcal{E}}, \hat{v}_{\mathcal{E}})^{\intercal}$. Then, Schwarz algorithm in the Fourier space reads

Given $\hat{\mathcal{E}}_i^n$, find $\hat{\mathcal{E}}_i^{n+1}$ such that

$$\begin{cases} \mathcal{L}(\hat{\underline{\mathcal{E}}}_1^{n+1}) = 0 \quad \text{with } \mathrm{x} < \delta, \\ \hat{\underline{\mathcal{E}}}_1^{n+1} \text{ bounded at infinity}, \\ \hat{\mathcal{T}}\hat{\underline{\mathcal{E}}}_1^{n+1}(\delta,k) = \hat{\mathcal{T}}\hat{\underline{\mathcal{E}}}_2^{n}(\delta,k),\ k \in \mathbb{R}, \end{cases} \qquad \begin{cases} \mathcal{L}(\hat{\underline{\mathcal{E}}}_2^{n+1}) = 0 \quad \text{with } \mathrm{x} > 0, \\ \hat{\underline{\mathcal{E}}}_2^{n+1} \text{ bounded at infinity}, \\ \hat{\mathcal{T}}\hat{\underline{\mathcal{E}}}_2^{n+1}(0,k) = \hat{\mathcal{T}}\hat{\underline{\mathcal{E}}}_1^{n}(0,k),\ k \in \mathbb{R}. \end{cases}$$

We get for a fixed k two ordinary differential equations in x, for each subdomain. Using maple software, we compute the solutions in each subdomain Ω_1 and Ω_2 The maple script is given in appendix A, the solutions read for $k > 0$

$$\begin{cases} \hat{u}_{\mathcal{E}}(x,k) = & a_1\exp(kx) + a_2 x\exp(kx) + a_3\exp(-kx) + a_4 x\exp(-kx) \\ \hat{v}_{\mathcal{E}}(x,k) = & -i\,\dfrac{[a_4\,k\,x\,(\lambda+\mu)\,\exp(-kx) - \,a_2\,k\lambda\,x\exp(kx)]}{k\,(\mu+\lambda)} \\ & -i\,\dfrac{[a_3\,k\,(\lambda+\mu)\,\exp(-kx) - a_2\,k\mu\,x\exp(kx)]}{k\,(\mu+\lambda)} \\ & +i\,\dfrac{[a_1\,k\,(\lambda+\mu)\,\exp(kx) + a_4\,(\lambda+3\mu)\,\exp(-kx) + a_2\,(\lambda+3\mu)\,\exp(kx)]}{k\,(\mu+\lambda)} \end{cases}$$

where a_1, a_2, a_3, a_4 are the integration constants. It is known that the solution must be bounded at infinity (in $-\infty$ for the subdomain Ω_1 and in $+\infty$ for the subdomain Ω_2), so we can reduce the solution to the following.

First, in Ω_1

$$\begin{cases} \hat{u}_{\mathcal{E}}(x,k) = & (a_1 + b_1 x)\exp(kx) \\ \\ \hat{v}_{\mathcal{E}}(x,k) = & \dfrac{i\left(a_1\mu k + b_1\mu kx + 3b_1\mu + a_1\lambda k + b_1\lambda kx + b_1\lambda\right)\exp(kx)}{k(\mu+\lambda)} \end{cases}$$

then in Ω_2

$$\begin{cases} \hat{u}_{\mathcal{E}}(x,k) = & (a_2 + b_2 x)\exp(-kx) \\ \\ \hat{v}_{\mathcal{E}}(x,k) = & \dfrac{i\left(a_2\mu k + b_2\mu kx + 3b_2\mu + a_2\lambda k + b_2\lambda kx + b_2\lambda\right)\exp(-kx)}{k(\mu+\lambda)} \end{cases}$$

a_1, b_1,a_2 and b_2 are the remained integration constants to be determined with the use of interface conditions.

3.3.2 Interface transmission conditions

Now depending on the choice of $\mathcal{T}$ operator, we get different interface conditions between subdomains. The best known are the Dirichlet interface condition, or the Neumann interface condition.

Dirichlet conditions impose the continuity of displacements (normal and tangential) in this case $\mathcal{T}$ corresponds to the identity operator, Neumann conditions impose the continuity of the normal stress vector. We recall that $\underline{\mathcal{E}}_i = (u_{\mathcal{E}i}, v_{\mathcal{E}i})$. $i = 1, 2$ corresponds to the vector field of displacement in each subdomain. Classical Dirichlet then Neumann interfaces read

$$(\mathbf{A}) \begin{cases} \underline{\mathcal{E}}_1^{n+1}(\delta, y) = \underline{\mathcal{E}}_2^{n}(\delta, y) \\ \\ \underline{\mathcal{E}}_2^{n+1}(0, y) = \underline{\mathcal{E}}_1^{n}(0, y) \end{cases} \qquad (\mathbf{B}) \begin{cases} \underline{\underline{\sigma}}_S(\underline{\mathcal{E}}_1^{n+1})(\delta, y).n_1 = \underline{\underline{\sigma}}_S(\underline{\mathcal{E}}_2^{n})(\delta, y).n_1 \\ \\ \underline{\underline{\sigma}}_S(\underline{\mathcal{E}}_2^{n+1})(0, y).n_2 = \underline{\underline{\sigma}}_S(\underline{\mathcal{E}}_1^{n})(0, y).n_2 \end{cases}$$

A mix of them can also be considered for the elasticity problem, it consists in imposing either continuity of the normal component of the normal stress vector and the tangential displacement or a continuity of the tangential component of the former and that of the normal component of the latter.

$$(\mathbf{C})\begin{cases} \underline{\underline{\sigma}}_S^N(\underline{\mathcal{E}}_1^{n+1})(\delta,y) = \underline{\underline{\sigma}}_S^N(\underline{\mathcal{E}}_2^{n})(\delta,y) \\ v_{\mathcal{E}1}^{n+1}(\delta,y) = v_{\mathcal{E}2}^{n}(\delta,y) \\ \underline{\underline{\sigma}}_S^N(\underline{\mathcal{E}}_2^{n+1})(0,y) = \underline{\underline{\sigma}}_S^N(\underline{\mathcal{E}}_1^{n})(0,y) \\ v_{\mathcal{E}2}^{n+1}(0,y) = v_{\mathcal{E}1}^{n}(0,y) \end{cases} \qquad (\mathbf{D})\begin{cases} u_{\mathcal{E}1}^{n+1}(\delta,y) = u_{\mathcal{E}2}^{n}(\delta,y) \\ \underline{\underline{\sigma}}_S^t(\underline{\mathcal{E}}_1^{n+1})(\delta,y) = \underline{\underline{\sigma}}_S^t(\underline{\mathcal{E}}_2^{n})(\delta,y) \\ u_{\mathcal{E}2}^{n+1}(0,y) = u_{\mathcal{E}1}^{n}(0,y) \\ \underline{\underline{\sigma}}_S^t(\underline{\mathcal{E}}_2^{n+1})(0,y) = \underline{\underline{\sigma}}_S^t(\underline{\mathcal{E}}_1^{n})(0,y) \end{cases}$$

where $\underline{\underline{\sigma}}_S^N$ and $\underline{\underline{\sigma}}_S^t$ are respectively the normal and the tangential components of the Cauchy stress tensor $\underline{\underline{\sigma}}_S.n$,

$$\underline{\underline{\sigma}}_S(\underline{\mathbf{u}}).n = \begin{bmatrix} \underline{\underline{\sigma}}_S^N \\ \\ \underline{\underline{\sigma}}_S^t \end{bmatrix} = \begin{bmatrix} \lambda\left(\dfrac{\partial u}{\partial x} + \dfrac{\partial v}{\partial y}\right) + 2\mu\dfrac{\partial u}{\partial x} \\ \mu\left(\dfrac{\partial v}{\partial x} + \dfrac{\partial u}{\partial y}\right) \end{bmatrix}$$

Thanks to the simple form of the solutions in the Fourier space, we can study the behavior of a_1, b_1 with respect to the iteration count n, then express the convergence factor of alternative Schwarz algorithm (3.8) & (3.9) for each interface condition type. We recall that a good algorithm is one for which the error $\underline{\hat{\mathcal{E}}}_i$ into the two subdomains with $i = 1, 2$ converges as quickly as possible to zero.

Using Maple software we can express the values a_1, b_1 through an iteration matrix that, when applied to the value (a_1, b_1) at iteration $n-1$ gives the value of (a_1, b_1) at iteration $n+1$

$$\begin{bmatrix} a_1^{n+1} \\ \\ b_1^{n+1} \end{bmatrix} = M_{it} \begin{bmatrix} a_1^{n-1} \\ \\ b_1^{n-1} \end{bmatrix} \tag{3.10}$$

Since the two subdomains Ω_1 and Ω_2 play a symmetric role, the changes in (a_2, b_2) in Ω_2 are described by the same iteration matrix M.

The resulting matrices depend heavily on the chosen interface conditions. In the two cases of mixed interface conditions (Neumann-Dirichlet) (**C**) and (**D**), the resulting matrix reads

$$M_{it} = \begin{bmatrix} e^{-2k\delta} & -2e^{-2k\delta} \\ \\ 0 & e^{-2k\delta} \end{bmatrix}.$$

For the other cases, the iteration matrices which are computed with Maple are a little more complicated. We will focus only on their two eigenvalues.

In the case (**A**) with Dirichlet interface conditions, we get

$$\begin{cases} eig_A[1] &= \left(1+2\dfrac{(\delta k)^2}{(3-4\nu)^2}+2\sqrt{\dfrac{(\delta k)^2}{(3-4\nu)^2}+\dfrac{(\delta k)^4}{(3-4\nu)^4}}\right)\exp(-2\,k\delta), \\ eig_A[2] &= \left(1+2\dfrac{(\delta k)^2}{(3-4\nu)^2}-2\sqrt{\dfrac{(\delta k)^2}{(3-4\nu)^2}+\dfrac{(\delta k)^4}{(3-4\nu)^4}}\right)\exp(-2\,k\delta). \end{cases} \tag{3.11}$$

In the case (**B**) with , Neumann interface conditions, we get

$$\begin{cases} eig_B[1] = \left(1+2\,\delta^2k^2+2\,\sqrt{\delta^2k^2+\delta^4k^4}\right)\exp(-2\,k\delta), \\ eig_B[2] = \left(1+2\,\delta^2k^2-2\,\sqrt{\delta^2k^2+\delta^4k^4}\right)\exp(-2\,k\delta). \end{cases} \tag{3.12}$$

And in the case (**C**) and (**D**), we get

$$eig_C[1] = eig_C[2] = eig_D[1] = eig_D[2] = \exp(-2k\delta) \tag{3.13}$$

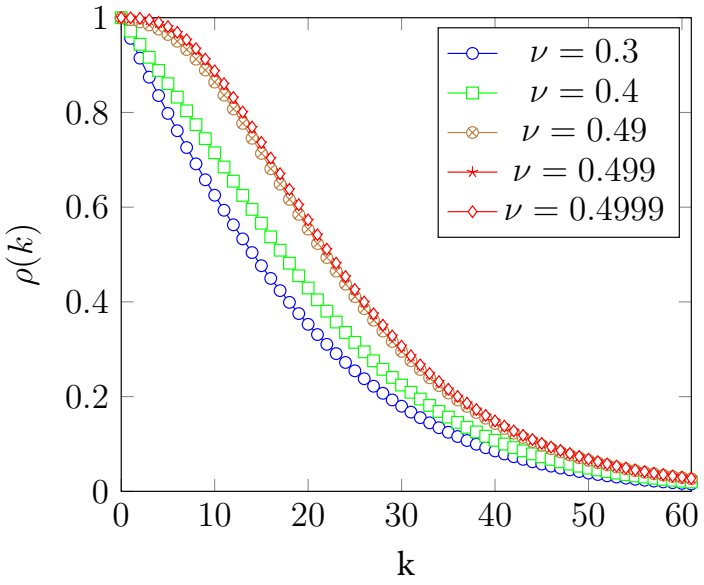


Fig 3.3 – Convergence factor vs Fourier number k for various Poisson's ratio, with Dirichlet interface condition, case:(**A**), overlap = 0.1.

3.3.3 Convergence factor

As it was studied earlier within the Ph.D thesis [Spi14], we recall some straightforward remarks

- In cases (**B**), (**C**) and (**D**), the eigenvalues do not depend on the material coefficient but they depend on the overlap δ and the frequency k
- From a comparison between eigenvalues of the several cases, we can make the following

observation

$$eig_A[1] > eig_C = eig_D > eig_A[2],$$

and

$$eig_B[1] > eig_C = eig_D > eig_B[2]$$

– Among the industrial engineering problems, we are interested in nearly incompressible materials. This is why we would like to study the behavior of the eigenvalues in the incompressible limit. This is relevant only in case (**A**) where interface conditions are in the pure displacement form. Lamé parameters can influence the convergence. We make also the following observation

$$\lim_{\nu \to 0.5} eig_A[1] = eig_B[1],$$

$$\lim_{\nu \to 0.5} eig_A[2] = eig_B[2].$$

The worst convergence is occurring at the incompressible limit. This is confirmed by the figure 3.3, where we plot the convergence as a function of the Fourier mode in the y direction for multiple Poisson's ratio going to the incompressible limit $\nu = 0.5$.

This study leads to the following conclusion: Schwarz Domain Decomposition method with Dirichlet or Neumann interface conditions are not suitable for elasticity equations applied to nearly incompressible materials. Mixed interface conditions performs better, but they are difficult to implement for arbitrary subdomains. For these reasons we made the choice to go to an other type of interface conditions in the next section.

3.3.4 Optimal Robin interface condition

Throughout the rest of this chapter we will consider a third type of interface condition. Robin (also called the Fourier conditions) are a weighted combination of Dirichlet and Neumann conditions. In this third case, $\mathcal{T}$ interface operator reads

$$\mathcal{T} := \underline{\underline{\sigma}}_S.n + \mathcal{S}. \tag{3.14}$$

Then for elasticity problem (3.6) in two dimension, interface conditions read

$$\begin{cases} \underline{\underline{\sigma}}_S(\underline{\mathcal{E}}_1^{n+1}).n_1 + \mathcal{S}_1\underline{\mathcal{E}}_1^{n+1}(\delta, k) = \underline{\underline{\sigma}}_S(\underline{\mathcal{E}}_2^{n}).n_1 + \mathcal{S}_1\underline{\mathcal{E}}_2^{n}(\delta, k) \\ \\ \underline{\underline{\sigma}}_S(\underline{\mathcal{E}}_2^{n+1}).n_2 + \mathcal{S}_2\underline{\mathcal{E}}_2^{n+1}(0, k) = \underline{\underline{\sigma}}_S(\underline{\mathcal{E}}_1^{n}).n_2 + \mathcal{S}_2\underline{\mathcal{E}}_1^{n}(0, k). \end{cases} \tag{3.15}$$

Returning to Fourier analysis, the Fourier transform in the y direction of Cauchy stress tensor $\underline{\underline{\sigma}}_S(\underline{\mathrm{u}}).n = [\underline{\underline{\sigma}}_S^t,\ \underline{\underline{\sigma}}_S^N]^\intercal$ is expressed as

$$\hat{\underline{\underline{\sigma}}}_S(\hat{\underline{\mathrm{u}}}).n = \begin{bmatrix} \hat{\underline{\underline{\sigma}}}_S^N \\ \\ \hat{\underline{\underline{\sigma}}}_S^t \end{bmatrix} = \begin{bmatrix} \lambda\left(\dfrac{\partial \hat{u}}{\partial x} + ik\dfrac{\partial \hat{v}}{\partial y}\right) + 2\mu\dfrac{\partial \hat{u}}{\partial x} \\ \mu\left(\dfrac{\partial \hat{v}}{\partial x} + ik\dfrac{\partial \hat{u}}{\partial y}\right) \end{bmatrix}.$$

Then Robin interface conditions in the 2D simplified equations read

$$\begin{cases} \hat{\underline{\underline{\sigma}}}_S^N(\hat{\underline{\mathcal{E}}}_1^{n+1}) + \hat{\mathcal{S}}_{1,1}\hat{u}_{\mathcal{E}1}^{n+1}(\delta,k) = \hat{\underline{\underline{\sigma}}}_S^N(\hat{\underline{\mathcal{E}}}_2^{n}) + \hat{\mathcal{S}}_{1,1}\hat{u}_{\mathcal{E}2}^{n}(\delta,k), & k\in\mathbb{R}, \\ \hat{\underline{\sigma}}_S^t(\hat{\underline{\mathcal{E}}}_1^{n+1}) + \hat{\mathcal{S}}_{1,2}\hat{v}_{\mathcal{E}1}^{n+1}(\delta,k) = \hat{\underline{\sigma}}_S^t(\hat{\underline{\mathcal{E}}}_2^{n}) + \hat{\mathcal{S}}_{1,2}\hat{v}_{\mathcal{E}2}^{n}(\delta,k), & k\in\mathbb{R}, \\ \hat{\underline{\underline{\sigma}}}_S^N(\hat{\underline{\mathcal{E}}}_2^{n+1}) + \hat{\mathcal{S}}_{2,1}\hat{u}_{\mathcal{E}2}^{n+1}(0,k) = \hat{\underline{\underline{\sigma}}}_S^N(\hat{\underline{\mathcal{E}}}_1^{n}) + \hat{\mathcal{S}}_{2,1}\hat{u}_{\mathcal{E}1}^{n}(0,k), & k\in\mathbb{R}, \\ \hat{\underline{\sigma}}_S^t(\hat{\underline{\mathcal{E}}}_2^{n+1}) + \hat{\mathcal{S}}_{2,2}\hat{v}_{\mathcal{E}2}^{n+1}(0,k) = \hat{\underline{\sigma}}_S^t(\hat{\underline{\mathcal{E}}}_1^{n}) + \hat{\mathcal{S}}_{2,2}\hat{v}_{\mathcal{E}1}^{n}(0,k), & k\in\mathbb{R}. \end{cases} \tag{3.16}$$

Coming back to the convergence factor of the Schwarz algorithm (3.8) and (3.9) where $\mathcal{T}$ is defined with the introduced Robin interface condition (3.15), a natural and crucial question is " Is there any optimal choice of $\mathcal{S}_i$? ". In order to give a response to this question we put the problem in the same framework as it has been treated in [HTJ88] then in [NRS94] for scalar operators. Among all the choices, there is one which is "theoretically" the best one since it provides a solution in a minimum number of iterations.

Indeed, we know that if an arbitrary initial guess is used, it is not possible to reach convergence in one iteration, but we can get a convergence in two iterations. For that it is required

$$\underline{\underline{\sigma}}_S(\underline{\mathcal{E}}_2^1).n_1 + \mathcal{S}_1\underline{\mathcal{E}}_2^1 = 0.$$

However, the only information available is related to $\underline{\mathcal{E}}_2^1$, which is that $\mathcal{L}(\underline{\mathcal{E}}_2^1) = 0$ in Ω_2 (i.e. $x > 0$). To exploit this only meaningful information, we introduce the DtN (Dirichlet-to-Neumann) operator, also called Steklov-Poincaré operator

$$\begin{aligned} &\text{for } u_0 : \Gamma_1 \longrightarrow \mathbb{R}^2 \\ &DtN_2(u_0) := \underline{\underline{\sigma}}_S(\underline{v}).n_{2|_{\partial\Omega_1\cap\overline{\Omega}_2}} \end{aligned}$$

where n_2 is the outward normal to $\Omega_2\backslash\overline{\Omega}_1$, and v satisfies the following boundary value problem

$$\begin{cases} \mathcal{L}(\underline{v}) = 0 & \Omega_2\backslash\overline{\Omega}_1 \\ \underline{v} = 0 & \partial\Omega_2\cap\partial\Omega \\ \underline{v} = \underline{u}_0 & \partial\Omega_1\cap\overline{\Omega}_2 \end{cases}.$$

If we chose

$$\mathcal{S}_1 := DtN_2,$$

since $n_1 = -n_2$, then we get

$$-\underline{\underline{\sigma}}_S(\underline{\mathcal{E}}_2^1).n_2 + DtN_2\underline{\mathcal{E}}_2^1 = 0\,.$$

Remark 6
Some remarks on the DtN operator

- The choice of $\mathcal{S}_i := DtN_i$ inside Robin interface condition is an optimal choice, it allows a convergence in two iterations

– This is a quite general result since we can see that it is completely independent from the problem operator $\mathcal{L}$.

In practice, we can compute the
operator in the case of elasticity problem in a simple geometry using the Fourier transform. We express the Robin interface conditions (3.15) depending on the constants integration a_1, a_2, a_3 and a_4. In our case the computation was done with maple software and the several steps are explained in detail in Appendix A.

In the 2D case we obtain the following expression of $\hat{\mathrm{DtN}}_2$ and $\hat{\mathrm{DtN}}_1$(resp. $\hat{\mathcal{S}}_1$ and $\hat{\mathcal{S}}_2$)

$$\hat{\mathcal{S}}_1 = \hat{\mathrm{DtN}}_2 = \begin{bmatrix} \dfrac{2\,|k|\,\mu(2\mu+\lambda)}{\lambda+3\mu} & \dfrac{2ik\mu^2}{\lambda+3\mu} \\ \dfrac{-2ik\mu^2}{\lambda+3\mu} & \dfrac{2\,|k|\,\mu(2\mu+\lambda)}{\lambda+3\mu} \end{bmatrix} \text{ and } \hat{\mathcal{S}}_2 = \hat{\mathrm{DtN}}_1 = \begin{bmatrix} \dfrac{2\,|k|\,\mu(2\mu+\lambda)}{\lambda+3\mu} & \dfrac{-2ik\mu^2}{\lambda+3\mu} \\ \dfrac{2ik\mu^2}{\lambda+3\mu} & \dfrac{2\,|k|\,\mu(2\mu+\lambda)}{\lambda+3\mu} \end{bmatrix}$$

. The action of DtN on $\underline{\mathrm{u}}_0$ is given by the following formula

$$\mathrm{DtN}_i\underline{\mathrm{u}}_0(\mathrm{x},\mathrm{y}) = \mathcal{F}^{-1}(\underline{\mathrm{u}}_0)(\mathrm{x},\mathrm{y}) = \frac{1}{2\pi}\int_{-\infty}^{+\infty} \hat{\mathcal{S}}_i\hat{\underline{\mathrm{u}}_0}\exp(iky)dk\,.$$

Remark 7
The latest result deserves few remarks. At this point DtN_i is a pseudo-differential operator whose symbol is $\hat{\mathcal{S}}_i$. Unfortunately, this choice is the best choice only from a theoretical point of view. These operators can be very difficult to implement in some cases, such as in presence of variable coefficient in the operator or in the case of curved boundary. It is almost impossible to obtain an exact form of the DtN_i operator.

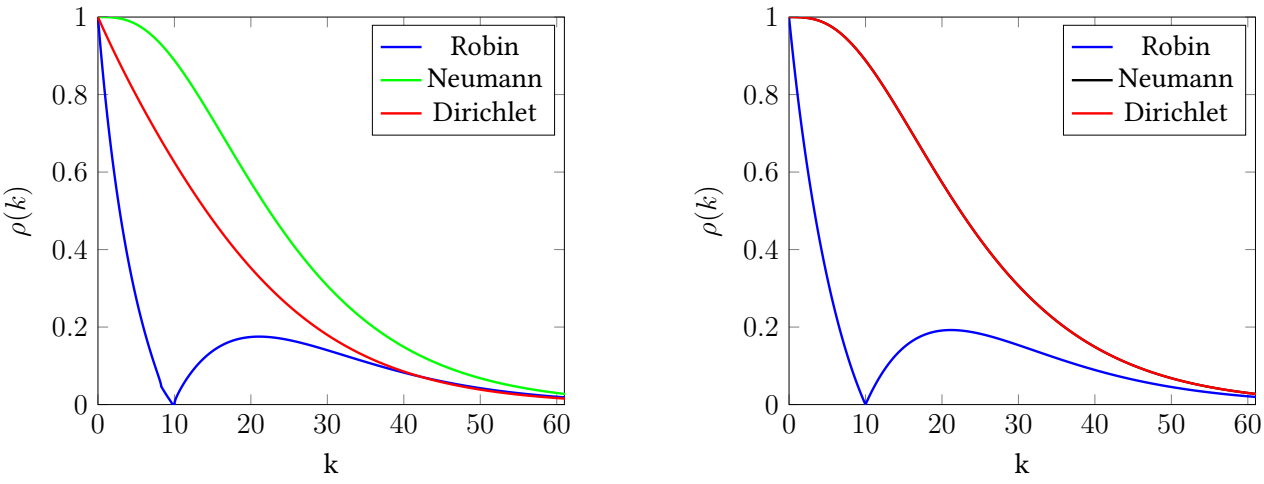
From DtN to Dtσ In fact we can use a part of the above ideally case, but with some changes. Indeed, we approximate the non differential entries of the DtN operators for some $k = k_0$. The approximated operator will be denoted Dtσ,

$$\hat{\mathcal{S}}_1 = \mathrm{Dt}\sigma_2 = \begin{bmatrix} \dfrac{2\,|k_0|\,\mu(2\mu+\lambda)}{\lambda+3\mu} & \dfrac{2ik\mu^2}{\lambda+3\mu} \\ \dfrac{-2ik\mu^2}{\lambda+3\mu} & \dfrac{2\,|k_0|\,\mu(2\mu+\lambda)}{\lambda+3\mu} \end{bmatrix}, \hat{\mathcal{S}}_2 = \mathrm{Dt}\sigma_1 = \begin{bmatrix} \dfrac{2\,|k_0|\,\mu(2\mu+\lambda)}{\lambda+3\mu} & \dfrac{-2ik\mu^2}{\lambda+3\mu} \\ \dfrac{2ik\mu^2}{\lambda+3\mu} & \dfrac{2\,|k_0|\,\mu(2\mu+\lambda)}{\lambda+3\mu} \end{bmatrix}$$

.

Hence, at this point we can write

$$\begin{aligned} \mathrm{DtN}_i\underline{\mathrm{u}}_0(\mathrm{x},\mathrm{y}) &= \frac{1}{2\pi}\int_{-\infty}^{+\infty} \hat{\mathcal{S}}_i\hat{\underline{\mathrm{u}}_0}(\mathrm{x},k)\exp(iky)dk \\ &= \mathcal{S}_i\frac{1}{2\pi}\int_{-\infty}^{+\infty} \hat{\underline{\mathrm{u}}_0}(\mathrm{x},k)\exp(iky)dk \\ &= \mathcal{S}_i\underline{\mathrm{u}}_0(\mathrm{x},\mathrm{y}) \end{aligned}$$



(a) Compressible case

(b) Incompressible case

Fig 3.4 – Convergence factor vs Fourier number k for various interface conditions(Dirichlet, Neumann and Robin), –Compressible material with $(E, \nu) = (210 \cdot 10^9, 0.3)$(left) and nearly-incompressible material with $(E, \nu) = (0.1 \cdot 10^9, 0.4999)$(left), overlap = 0.1.

If we do not take into account the boundary terms, the operator is symmetric, but we will see in the next section that the following further simplified version works fine from the convergence factor point of view.

$$\hat{\mathcal{S}}_1 = \mathrm{Dt}\sigma_2 = \begin{bmatrix} \frac{2\,|k_0|\,\mu(2\mu+\lambda)}{\lambda+3\mu} & 0 \\ 0 & \frac{2\,|k_0|\,\mu(2\mu+\lambda)}{\lambda+3\mu} \end{bmatrix}, \hat{\mathcal{S}}_2 = \mathrm{Dt}\sigma_1 = \begin{bmatrix} \frac{2\,|k_0|\,\mu(2\mu+\lambda)}{\lambda+3\mu} & 0 \\ 0 & \frac{2\,|k_0|\,\mu(2\mu+\lambda)}{\lambda+3\mu} \end{bmatrix}$$

3.3.5 Convergence factor

In figure 3.4, we plot the convergence as a function of the Fourier mode in the y direction, first for elasticity problem on a compressible material such as steel with Lamé coefficients $(E, \nu) = (210 \cdot 10^9, 0.3)$. In the second case (b) we plot the convergence for a nearly-incompressible material such as rubber with Lamé coefficient $(E, \nu) = (0.1 \cdot 10^9, 0.4999)$, the overlap is of size $\delta = 0.1$.

We can observe that the approximation of the DtN operator has an impact on the convergence factor of the Optimized Schwarz algorithm, see Figure 3.4. Although the Robin interface condition using the approximated DtN (3.16) is never exact even for $k = k_0$ ($k_0 = 10$ here), the convergence factor is quite close to zero (of the order of 10^{-4}) for $k = k_0$. Note that Dirichlet or Neumann (stress free) interface conditions yield the same convergence factor in case (b). For small Fourier numbers, the convergence factor is very close to 1 which is bad. Overall, Robin interface conditions perform much better than simple Dirichlet or Neumann interface conditions.

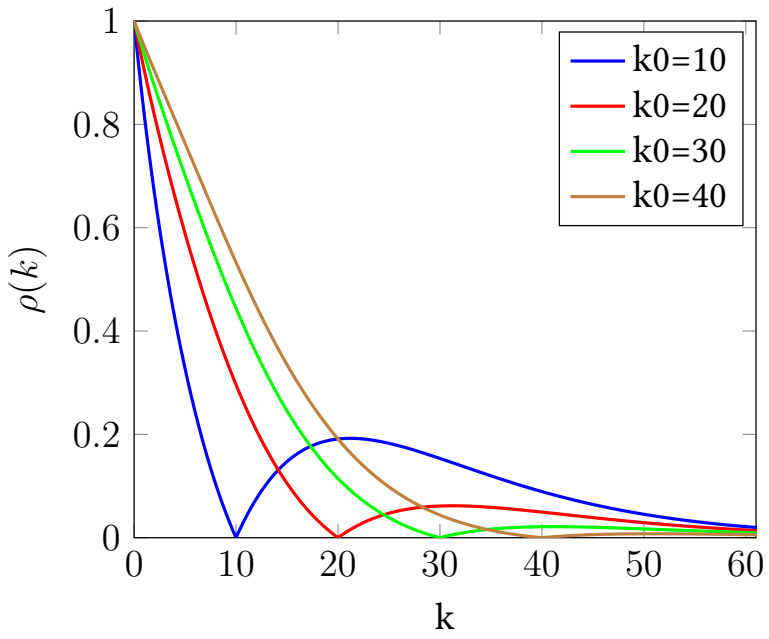


Fig 3.5 – Convergence factor vs Fourier number k for Robin interface condition with various k_0. nearly-incompressible material with $(E, \nu) = (0.1 \cdot 10^9, 0.4999)$(left), overlap = 0.1.

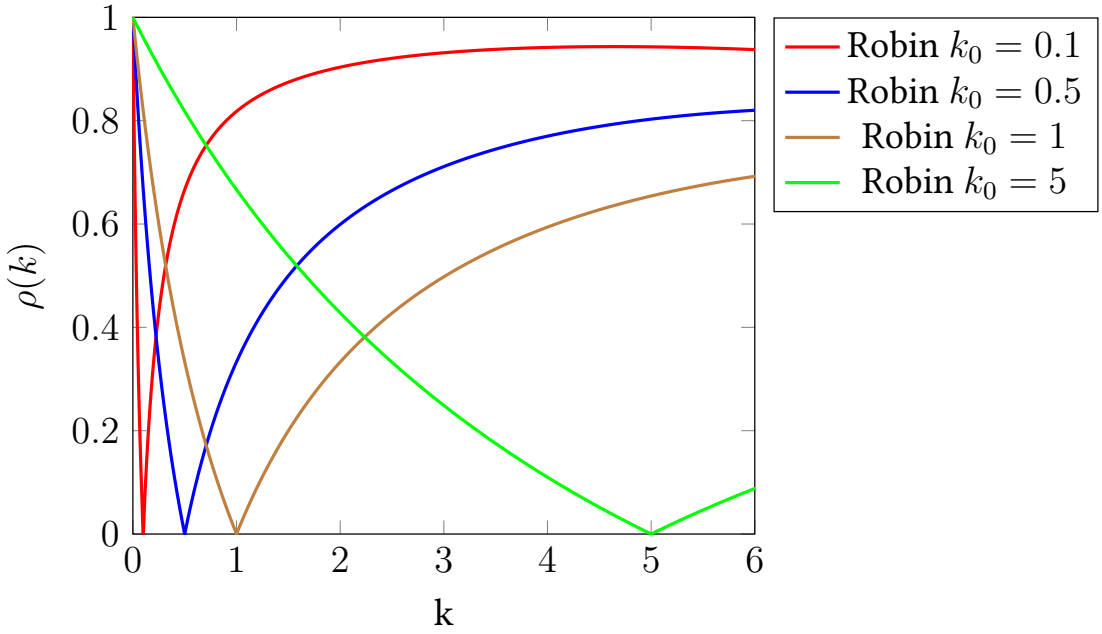


Fig 3.6 – Other view on the convergence factor vs Fourier number k for Robin interface condition with **"small"** k_0. nearly-incompressible material with $(E, \nu) = (0.1 \cdot 10^9, 0.4999)$, overlap = 0.1.

3.4 Stokes equations

A similar Fourier analysis can be conducted for the following Stokes equation

$$\begin{cases} \mathcal{L}(\underline{\mathbf{u}}, p) := \underline{\nabla} \cdot \underline{\underline{\sigma}}_F(\underline{\mathbf{u}}, p) & = \underline{\mathbf{f}} \text{ in } \Omega \\ \underline{\mathbf{u}}(\mathbf{x}) \text{ bounded at infinity,} \end{cases} \tag{3.17}$$

where

$$\underline{\underline{\sigma}}_F = -pI + 2\mu\underline{\underline{\varepsilon}}(\underline{\mathbf{u}}) \quad \text{and} \quad \underline{\underline{\varepsilon}}(\underline{\mathbf{u}}) = \frac{1}{2}\left(\underline{\underline{\nabla}}\,\underline{\mathbf{u}} + (\underline{\underline{\nabla}}\,\underline{\mathbf{u}})^T\right).$$

Then with

$$\underline{\nabla} \cdot \underline{\underline{\sigma}}_F(\underline{\mathbf{u}}, p) = \begin{bmatrix} -\partial_x p + 2\mu\partial_{xx}u + \mu\partial_{xy}v + \mu\partial_{yy}u \\ \\ 2\mu\partial_{yy}v - \partial_y p + \mu\partial_{xy}u + \mu\partial_{xx}v \end{bmatrix},$$

Stokes equation in a 2D framework reads

$$\mathcal{L}(\underline{\mathbf{u}}, p) = \begin{cases} \partial_x p - \mu\Delta u - \mu\partial_x \underline{\nabla} \cdot (\underline{\mathbf{u}}) = f_1 \\ \partial_y p - \mu\Delta v - \mu\partial_y \underline{\nabla} \cdot (\underline{\mathbf{u}}) = f_2 \end{cases}, \text{ with } \underline{\mathbf{u}} = (u, v)$$

Stokes formulation is close to elasticity formulation, by analogy we can observe that the coefficient μ in Lamé coefficients becomes the kinematic viscosity in Stokes equation and λ will tend to infinity. This leads us to use the same approximated DtN (i.e Dtσ) operator as for elasticity equation with a high λ.

3.5 From Lions algorithm to ORAS algorithm

The objective of this section is to present the algebraic formulation of Lions' algorithm which is the ORAS algorithm. The ORAS algorithm was defined in [SCGT07a] at the algebraic level. We will not follow the chronological order but rather proceed the other way around. We introduce first a continuous version of ORAS (Optimized Restrictive Additive Schwarz) algorithm which will allows us to move towards a purely discrete version of Lions' algorithm.

Let $\Omega \subset \mathbb{R}^N$ be a bounded smooth domain, we consider an overlapping decomposition into 2 subdomains of Ω such that

$$\Omega := \Omega_1 \cup \Omega_2 \text{ with} \quad \Gamma_{12} = \partial\Omega_1 \cap \overline{\Omega}_2, \quad \Gamma_{21} = \partial\Omega_2 \cap \overline{\Omega}_1.$$

Γ_{12} (resp. Γ_{21}, see figure 3.2) is the interface between subdomain Ω_1 and Ω_2. In general case, for a linear PDE operator $\mathcal{L}$ which acts on functions and a given source term f. Schwarz algorithm with Robin interface condition (i.e. Lions' algorithm) reads

Given arbitrary initial guesses $(\mathrm{u}_i^0)_{1\leqslant i\leqslant 2}$, find $(\mathrm{u}_i^{n+1})_{1\leqslant i\leqslant 2}$ such that

$$\begin{cases} \mathcal{L}(\mathrm{u}_i^{n+1}) = \mathrm{f} & \text{in } \Omega_i \\ \mathrm{u}_i^{n+1} = 0 & \text{on } \partial\Omega_i \cap \partial\Omega \\ \left(\dfrac{\partial}{\partial n_{ij}} + \lambda_{ij}\right)\mathrm{u}_i^{n+1} = \left(\dfrac{\partial}{\partial n_{ij}} + \lambda_{ij}\right)\mathrm{u}_j^{n} & \text{on } \Gamma_{ij} = \partial\Omega_i \cap \overline{\Omega}_j \end{cases} \tag{3.18}$$

In order to introduce ORAS (Optimized Restrictive Additive Schwarz) algorithm, we recall the definition of some ingredients.
First of all, we provide the overlapping decomposition $\{\Omega_i\}_{1\leqslant i\leqslant 2}$ of the domain Ω with a partition of unity $(\chi_i)_{1\leqslant i\leqslant 2}$ and extension operators $(\mathrm{E}_i)_{1\leqslant i\leqslant 2}$.

Definition 10 (Extension & Partition of unity operators)
.

- ***Extension & Restriction:*** *Let E_i be an extension operator defined as $E_i(\omega_i) : \Omega \longrightarrow \mathbb{R}$ such that E_i is the extension function of $\omega_i : \Omega_i \longrightarrow \mathbb{R}$ by zero outside Ω_i.*
- ***Partition of unity:*** *Let χ_i be the partition of unity $\chi_i : \Omega_i \longrightarrow \mathbb{R}$, such that $\chi \geqslant 0$, $\chi_i(\mathrm{x}) = 0$ on a neighborhood of $\partial\Omega_i \backslash \partial\Omega$ and*

$$\omega = \sum_{i=1}^{2} E_i(\chi_i \omega_{|\Omega_i})$$

for all functions $\omega : \Omega \longrightarrow \mathbb{R}$

Then, the continuous ORAS algorithm can be written as in Algorithm 2.

Algorithm 2 continuous ORAS algorithm

1: Compute residual
$$r^n = \mathrm{f} - \mathcal{L}(\mathrm{u}^n)$$
2: **for** each subdomain $i = 1, 2$, compute a local correction **do**
$$\begin{cases} \mathcal{L}(\mathrm{v}_i^{n+1}) = r^n & \text{in } \Omega_i \\ \mathrm{v}_i^{n+1} = 0 & \text{on } \partial\Omega_i \cap \partial\Omega \\ \left(\dfrac{\partial}{\partial n_{ij}} + \lambda_{ij}\right)\mathrm{v}_i^{n+1} = 0 & \text{on } \Gamma_{ij} = \partial\Omega_i \cap \overline{\Omega}_j \end{cases}$$
3: **end for**
4: then update u^n by
$$\mathrm{u}^{n+1} = \mathrm{u}^n + \mathrm{E}(\chi_1 \mathrm{v}_1^{n+1}) + \mathrm{E}(\chi_2 \mathrm{v}_2^{n+1})$$
where E_i and χ_i are respectively the extension operators and the partition of unity defined in 10

Similarly to the established equivalence between Schwarz and RAS algorithm in Chapter 2, it is also possible to have an equivalence between Lions' algorithm and continuous ORAS Algorithm (Algorithm 2). Notice that the latter algorithm iterates on the global function $\mathrm{u}^n : \Omega \longrightarrow \mathbb{R}$.

Lemma 11
Continuous ORAS algorithm given by Algorithm 2 is equivalent to Lions' algorithm defined in (3.18).

Proof. In the same way that it was done for the equivalence between Schwarz algorithm and RAS algorithm, by hypothesis, we know that for the initial guess

$$\mathrm{u}^0 = \mathrm{E}_1(\chi_1 \mathrm{u}_1^0) + \mathrm{E}(\chi_2 \mathrm{u}_2^0).$$

The main point to prove is the following equality for every n

$$\mathrm{u}^n = \mathrm{E}_1(\chi_1 \mathrm{u}_1^n) + \mathrm{E}_2(\chi_2 \mathrm{u}_2^n),$$

where $\mathrm{u}_1^n, \mathrm{u}_2^n$ are given by Lions algorithm in (3.18) and u^n is a solution given by ORAS algorithm 2. To do this, we proceed by induction. We assume that the property holds at the nth iteration of the algorithms

$$\mathrm{u}^n = \mathrm{E}(\chi_1 \mathrm{u}_1^n) + \mathrm{E}(\chi_2 \mathrm{u}_2^n)\,.$$

From (3.18), we have

$$\mathrm{u}^{n+1} = \mathrm{E}(\chi_1(\mathrm{u}_{|\Omega_1}^n + \mathrm{v}_1^n)) + \mathrm{E}(\chi_2(\mathrm{u}_{|\Omega_2}^n + \mathrm{v}_2^n)).$$

Now we prove that

$$\mathrm{u}_1^{n+1} = \mathrm{u}_{|\Omega_1}^n + \mathrm{v}_1^n.$$

This requires to prove that $\mathrm{u}_{|\Omega_1}^n + \mathrm{v}_1^n$ is a solution to the problem (3.18)

$$\begin{cases} \mathcal{L}(\mathrm{u}_{|\Omega_1}^n + \mathrm{v}_1^n) = \mathcal{L}(\mathrm{u}_{|\Omega_1}^n) + r^n = \mathcal{L}(\mathrm{u}_{|\Omega_1}^n) + f - \mathcal{L}(\mathrm{u}_{|\Omega_1}^n) = \mathrm{f} \quad \text{in } \Omega_1 \\ \mathrm{u}_{|\Omega_1}^n + \mathrm{v}_1^n = 0 \quad \text{on } \partial\Omega_1 \cap \partial\Omega \\ \left(\dfrac{\partial}{\partial n_{1j}} + \lambda_{1j}\right)(\mathrm{u}_{|\Omega_1}^n + \mathrm{v}_1^n) = \left(\dfrac{\partial}{\partial n_{1j}} + \lambda_{1j}\right)\mathrm{u}_{|\Omega_1}^n \quad \text{on } \Gamma_{1j} \quad \forall j, \quad 1 \leqslant j \leqslant 2 \quad j \neq 1 \end{cases}$$

The last thing to prove is

$$\left(\frac{\partial}{\partial n_{1j}} + \lambda_{1j}\right)\mathrm{u}_{|\Omega_1}^n = \left(\frac{\partial}{\partial n_{1j}} + \lambda_{1j}\right)\mathrm{u}_2^n\,.$$

From the induction hypothesis $\mathrm{u}^n = \mathrm{E}(\chi_1 \mathrm{u}_1^n) + \mathrm{E}(\chi_2 \mathrm{u}_2^n)$, on a neighborhood of the interface Γ_{12} we have $\chi_1 \equiv 0$ and $\chi_2 \equiv 1$. Thus on Γ_{12} we have

$$\left(\frac{\partial}{\partial n_{1j}} + \lambda_{1j}\right)\mathrm{u}_{|\Omega_1}^n = \left(\frac{\partial}{\partial n_{1j}} + \lambda_{1j}\right)(\chi_1 \mathrm{u}_1^n + \chi_2 \mathrm{u}_2^n) = \left(\frac{\partial}{\partial n_{1j}} + \lambda_{1j}\right)\mathrm{u}_2^n.$$

Hence, we can say that $\mathrm{u}_{|\Omega_1}^n + \mathrm{v}_1^n = \mathrm{u}_1^{n+1}$ which satisfies (3.18). The same proof holds for Ω_2, where $\mathrm{u}_{|\Omega_2}^n + \mathrm{v}_2^n = \mathrm{u}_2^{n+1}$. Finally this allows us to write

$$\mathrm{u}^{n+1} = \mathrm{E}_1(\chi_1 \mathrm{u}_1^{n+1}) + \mathrm{E}_2(\chi_2 \mathrm{u}_2^{n+1}).$$

This concludes the proof of equivalence between P.L Lions algorithm and continuous ORAS

algorithm. ■

3.6 ORAS algorithm

In this section, we write the algebraic formulation of the continuous ORAS Algorithm 3. As it was done in the chapter 2 for RAS algorithm.

First, we recall that we are interested in solving a PDE problem given by an operator $\mathcal{L}$ and boundary conditions defined in the domain Ω. In the case of Galerkin methods we are led to solve the following problem

$$\text{Find } u \in V \text{such that : } a_\Omega(u,v) = l(v)\,, \quad \forall v \in V\,,$$

where V is a Hilbert space of functions. We consider a finite element discretization. Let $\mathcal{T}_h$ be the triangulation of the domain Ω, $\mathcal{N}$ denote the set of degrees of freedom and $(\phi_k)_{k\in\mathcal{N}}$ be a finite element basis on a mesh $\mathcal{T}_h$. Let $A \in \mathbb{R}^{\#\mathcal{N}\times\#\mathcal{N}}$ be the associated finite element matrix, $A_{kl} := a_\Omega(\phi_l,\phi_k)$, $k,l \in \mathcal{N}$. For some given right hand side $\mathbf{F} \in \mathbb{R}^{\#\mathcal{N}}$, we have to solve a linear system in $\mathbf{U}$ of the form

$$A\mathbf{U} = \mathbf{F}\,.$$

In order to write the ORAS algorithm, we need to recall the discrete counterpart of extension and partition of unity operators.

- Let Ω (resp. $\mathcal{T}_h$) be decomposed into N subdomains $(\Omega_i)_{1\leq i\leq N}$ (resp. $(\mathcal{T}_{i,h})_{1\leq i\leq N}$) so that all subdomains are a union of cells of the mesh $\mathcal{T}_h$.

$$\Omega := \bigcup_{i=1}^{N}\{\overline{\Omega}_i\} = \bigcup_{i=1}^{N}\mathcal{T}_{i,h} \quad \text{and} \quad \Omega_i := \bigcup_{\tau\in\mathcal{T}_{i,h}}\tau \quad \text{for } 1\leqslant i\leqslant N$$

- This decomposition induces a natural decomposition of the set of indices $\mathcal{N}$ into N subsets of indices $(\mathcal{N}_i)_{1\leq i\leq N}$:

$$\mathcal{N} := \bigcup_{i=1}^{m}\mathcal{N}_i.$$

- In the case of a non-overlapping decomposition, we extend each subdomain Ω_i to a domain Ω_i^δ by adding one or several layers (which can be adjacent fine grid, see figure 2.3), this will induce the extension of each subset $\mathcal{N}_i$ with its direct neighbors, which will form $\mathcal{N}_i^\delta$, δ is the size of overlap.

- We consider a vector $\mathbf{U} \in \mathbb{R}^{\#\mathcal{N}}$, then for all $1 \leq i \leq N$, let R_i be a matrix from $\mathbb{R}^{\#\mathcal{N}}$ to the subset $\mathbb{R}^{\#\mathcal{N}_i^\delta}$. The algebraic counterpart part of restriction operator is defined in Definition 10

- Let D_i be a diagonal matrix of size $\#\mathcal{N}_i^\delta \times \#\mathcal{N}_i^\delta$. The algebraic counterpart part of partition of unity operator defined in definition 4. see figure 2.2. So, that we have a partition of

unity at the algebraic level,

$$\sum_{i=1}^{N} R_i^T D_i R_i = I_d \,,$$

where $I_d \in \mathbb{R}^{\#\mathcal{N}\times\#\mathcal{N}}$ is the identity matrix.

– For all subdomains $1 \leq i \leq N$, let B_i be a local discretization matrix of size $\#\mathcal{N}_i^\delta \times \#\mathcal{N}_i^\delta$, which comes typically from the discretization of boundary value local problems with Robin as boundary condition in subdomain interface.

Definition 12 (discrete ORAS algorithm)
The Optimized Restrictive Schwarz algorithm (ORAS, see [SCGT07a]) is defined as a preconditioner for the following fixed-point problem

$$\mathbf{U}^{n+1} = \mathbf{U}^n + M_{ORAS,1}^{-1}(\mathbf{F} - A\mathbf{U}^n)$$

with

$$M_{ORAS,1}^{-1} := \sum_{i=1}^{N} R_i^T D_i B_i^{-1} R_i \,. \tag{3.19}$$

Algorithm 3 Algebraic ORAS algorithm

1: Compute residual $r^n \in \mathbb{R}^{\#\mathcal{N}}$

$$r^n = \mathbf{F} - A\mathbf{U}^n$$

2: **for** each subdomain $i = 1, \quad 1 \leqslant i \leqslant N$ to compute a local correction $\mathbf{V}_i^{n+1}$ **do**

$$B_{i,\text{Robin}}\mathbf{V}_i^{n+1} = R_i r^n$$

where B_i^{Robin} is the discretization matrix of local Robin problem .

3: **end for**

4: Compute an approximation for local correction then update $\mathbf{U}^n$ by

$$\mathbf{U}^{n+1} = \mathbf{U}^n + \sum_{i=1}^{m} R_i^T D_i B_{i,\text{Robin}}^{-1} R_i \mathbf{V}^{n+1}.$$

where R_i and D_i are respectively the algebraic counterpart of extension operators and the partition of unity defined in 10

Remark 8
Formula (3.19) differs from the definition of the RAS algorithm only in the local matrix that has to be inverted.

Chapter 4

Two level SORAS-GenEO-2

Contents

4.1 Introduction

Substructuring algorithms such as BNN (Balancing Neumann-Neumann) or FETI (finite element tearing and interconnecting) are defined for nonoverlapping domain decompositions but not for overlapping subdomains. Schwarz method [Sch70] is defined only for overlapping subdomains. With the help of a coarse space correction, the two-level versions of both type of methods are weakly scalable, see [TW05] and references therein. The domain decomposition method introduced by P.L. Lions [Lio90] is a third type of methods. It can be applied to both overlapping and nonoverlapping subdomains. It is based on improving Schwarz methods by replacing the Dirichlet interface conditions by Robin interface conditions. This algorithm was extended to

Helmholtz problem by Després [Des93]. Robin interface conditions can be replaced by more general interface conditions that can be optimized (Optimized Schwarz methods, OSM) for a better convergence, see [GMN02; Gan06] and references therein.

P.L. Lions proved the convergence of his algorithm in the elliptic case for a nonoverlapping domain decomposition. The proof is based on energy estimates and a summation technique. These results were extended to Helmholtz and Maxwell equations in [BD97; DJR92]. Over the last years, a lot of results have been obtained for different classes of equations and optimized algorithms based on carefully chosen parameters in the transmission conditions, have been derived, see e.g. [JNR98; Gan06; GMN02; DGG09] and references therein. Most of these works are valid for nonoverlapping decomposition or for simple overlapping domain decompositions. When the domain is decomposed into a large number of subdomains, these methods are, on a practical point of view, scalable if a second level is added to the algorithm via the introduction of a coarse space [JNR98; FML00; CDKN14]. But there is no systematic procedure to build coarse spaces with a provable efficiency.

The purpose of this chapter is to define a general framework for building adaptive coarse space for OSM methods for decomposition into overlapping subdomains. We prove that we can achieve the same robustness that what was done for Schwarz [SDHNPS14] and FETI-BDD [SDHNR13] domain decomposition methods with so called GenEO (Generalized Eigenvalue in the Overlap) coarse spaces. Compared to these previous works, we have to introduce SORAS (symmetrized ORAS) a non standard symmetric variant of the ORAS method as well as two generalized eigenvalue problems. As numerical results will show in section 4.7, the method scales very well for saddle point problems such as highly heterogeneous nearly incompressible elasticity problems as well as the Stokes system. More precisely, in section 4.2, we give a short presentation of the current theory for the additive Schwarz method. Then, in section 4.3, we present algebraic variants to the P.L. Lions' domain decomposition method. In section 4.4, we build a coarse space so that the two-level SORAS method achieves a targeted condition number. In section 4.5, the method is applied to saddle point problems.

4.2 Short introduction to ASM theory

In order to appraise the theory developed in § 4.3, we first give a short presentation of the current theory for two-level additive Schwarz methods. The starting point was the original Schwarz algorithm [Sch70] for proving the well-posedness of the Poisson problem $-\Delta u = f$ with Dirichlet boundary conditions in some domain Ω decomposed into two subdomains Ω_1 and Ω_2, $\Omega = \Omega_1 \cup \Omega_2$.

Definition 13 (Alternating Schwarz algorithm)
The Schwarz algorithm is an iterative method based on solving alternatively sub-problems in domains Ω_1 and Ω_2. It updates $(\mathrm{u}_1^n, \mathrm{u}_2^n) \to (\mathrm{u}_1^{n+1}, \mathrm{u}_2^{n+1})$ by:

$$\begin{cases} -\Delta(\mathrm{u}_1^{n+1}) = & \mathrm{f} \text{ in } \Omega_1 \\ \mathrm{u}_1^{n+1} = & 0 \text{ on } \partial\Omega_1 \cap \partial\Omega \\ \mathrm{u}_1^{n+1} = & \mathrm{u}_2^n \text{ on } \partial\Omega_1 \cap \overline{\Omega_2}. \end{cases} \text{ then, } \begin{cases} -\Delta(\mathrm{u}_2^{n+1}) = & \mathrm{f} \text{ in } \Omega_2 \\ \mathrm{u}_2^{n+1} = & 0 \text{ on } \partial\Omega_2 \cap \partial\Omega \\ \mathrm{u}_2^{n+1} = & \mathrm{u}_1^{n+1} \text{ on } \partial\Omega_2 \cap \overline{\Omega_1}. \end{cases} \tag{4.1}$$

H. Schwarz proved the convergence of the algorithm and thus the well-posedness of the Poisson problem in complex geometries. A small modification of the algorithm [Lio90] makes

it suited to parallel architectures. Its convergence can be proved using the maximum principle [Lio89].

Definition 14 (Parallel Schwarz algorithm)
Iterative method which solves concurrently in all subdomains, $i = 1, 2$*:*

$$\begin{cases} -\Delta(\mathrm{u}_i^{n+1}) = \mathrm{f} & \text{in} \quad \Omega_i \\ \mathrm{u}_i^{n+1} = 0 & \text{on} \quad \partial\Omega_i \cap \partial\Omega \\ \mathrm{u}_i^{n+1} = \mathrm{u}_{3-i}^{n} & \text{on} \quad \partial\Omega_i \cap \overline{\Omega}_{3-i}. \end{cases} \tag{4.2}$$

The discretization of this algorithm yields a parallel algebraic method for solving the linear system $A\mathbf{U} = \mathbf{F} \in \mathbb{R}^{\#\mathcal{N}}$ ($\mathcal{N}$ is the set of degrees of freedom) arising from the discretization of the original Poisson problem set on domain Ω. Due to the duplication of the unknowns in the overlapping region $\Omega_1 \cap \Omega_2$, this direct discretization involves a matrix of size larger than that of matrix A, see e.g. [HMNSXZ13] for more details. Actually, it is much simpler and as efficient to use the RAS preconditioner [CS99]

$$M_{RAS}^{-1} := \sum_{i=1}^{N} R_i^T D_i A_i^{-1} R_i \,, \tag{4.3}$$

where N is the number of subdomains, R_i for some $1 \leq i \leq N$ is the Boolean matrix that restricts a global vector to its degrees of freedom in subdomain Ω_i, matrix

$$A_i := R_i A_i R_i^T$$

is the Dirichlet matrix of subdomain Ω_i and D_i is a local diagonal matrix that yields an algebraic partition of unity on $\mathbb{R}^{\#\mathcal{N}}$:

$$I_d = \sum_{i=1}^{N} R_i^T D_i R_i \,. \tag{4.4}$$

Indeed, it is proved in [EG03] that the following fixed point algorithm

$$\mathbf{U}^{n+1} = \mathbf{U}^n + M_{RAS}^{-1}(\mathbf{F} - A\mathbf{U}^n) \tag{4.5}$$

yields iterates that are equivalent to that of the discretization of Algorithm (4.2).

The RAS preconditioner (4.3) is not symmetric. In order to develop a theory for it when used as preconditioner in a Krylov method, its symmetric variant, the ASM preconditioner:

$$M_{ASM}^{-1} = \sum_{i=1}^{N} R_i^T A_i^{-1} R_i \,, \tag{4.6}$$

was studied extensively, see [TW05] and references therein. Starting with the pioneering work [Nic87], a lot of effort has been devoted to the design and analysis of two-level methods that are the key ingredient to scalable methods. In adaptive methods, the coarse space in the two-level method is built by solving local generalized eigenvalue problems [GE10; EGLW12; NXDS11; SDHNPS14]. This way, it is possible to target a user defined condition number of the preconditioned system. Here we focus on the GenEO approach [SDHNPS14] where the coarse space is based on

solving Generalized Eigenvalue problems for the set of degrees of freedom $\mathcal{N}_j$ of subdomain $1 \leq j \leq N$. Let A_j^{Neu} denote the matrix of the local Neumann problem, we have to find the eigenpairs $(V_{j,k}, \lambda_{j,k})_k$ such that:
$V_{j,k} \in \mathbb{R}^{\mathcal{N}_j}$ and $\lambda_{j,k} \geq 0$:

$$D_j\, A_j D_j V_{j,k} = \lambda_{j,k}\, A_j^{Neu} V_{j,k} \tag{4.7}$$

By combining the eigenvectors corresponding to eigenvalues larger than some given threshold $\tau > 0$ into a coarse space, it is proved in [SDHNPS14; DJN15] that the eigenvalues of the hybrid Schwarz preconditioned system satisfy the following estimate

$$\frac{1}{1 + k_1\, \tau} \leq \lambda(M_{HSM}^{-1}\, A) \leq k_0\,. \tag{4.8}$$

where k_0 is the maximum number of neighbors of a subdomain and k_1 is the maximum multiplicity of the intersections of subdomains.

To sum up, the current theory for the two-level Schwarz method is based on the following four steps:

1. Schwarz algorithm at the continuous level (4.1)
2. An equivalent algebraic formulation (4.5) with the introduction of the RAS preconditioner (4.3)
3. A symmetrized variant named ASM (4.6) of the RAS preconditioner
4. A two-level method with an adaptive coarse space with prescribed targeted convergence rate .

4.3 Symmetrized ORAS method

Our goal here is to develop a theory and computational framework for P.L. Lions algorithm similar to what was done for the Schwarz algorithm for a SPD matrix A. We follow the steps recalled above.

First we introduce the P.L. Lions' algorithm which is based on improving Schwarz methods by replacing the Dirichlet interface conditions by Robin interface conditions. Let α be a positive number, the modified algorithm reads

$$\left\{\begin{array}{rcll} -\Delta(\mathrm{u}_1^{n+1}) & = & f & \text{in} \quad \Omega_1, \\ \mathrm{u}_1^{n+1} & = & 0 & \text{on} \quad \partial\Omega_1 \cap \partial\Omega, \\ \left(\dfrac{\partial}{\partial \mathbf{n}_1} + \alpha\right)(\mathrm{u}_1^{n+1}) & = & \left(\dfrac{\partial}{\partial \mathbf{n}_1} + \alpha\right)(\mathrm{u}_2^{n}) & \text{on} \quad \partial\Omega_1 \cap \overline{\Omega_2}\,, \end{array}\right. \tag{4.9}$$

and

$$\left\{\begin{array}{rcll} -\Delta(\mathrm{u}_2^{n+1}) & = & f & \text{in} \quad \Omega_2, \\ \mathrm{u}_2^{n+1} & = & 0 & \text{on} \quad \partial\Omega_2 \cap \partial\Omega \\ \left(\dfrac{\partial}{\partial \mathbf{n}_2} + \alpha\right)(\mathrm{u}_2^{n+1}) & = & \left(\dfrac{\partial}{\partial \mathbf{n}_2} + \alpha\right)(\mathrm{u}_1^{n}) & \text{on} \quad \partial\Omega_2 \cap \overline{\Omega_1} \end{array}\right. \tag{4.10}$$

where $\mathbf{n}_1$ and $\mathbf{n}_2$ are the outward normals on the boundary of the subdomains.

The second step is an algebraic equivalent formulation of the P.L. Lions algorithm in the case of overlapping subdomains. It is based on the introduction of the ORAS (Optimized Restricted Additive Schwarz) [SCGT07b] preconditioner:

$$M_{ORAS}^{-1} := \sum_{i=1}^{N} R_i^T D_i B_i^{-1} R_i \,, \tag{4.11}$$

where $(B_i)_{1\leq i\leq N}$ is the discretization matrix of the Robin problem in subdomain Ω_i. The following fixed point method

$$\mathbf{U}^{n+1} = \mathbf{U}^n + M_{ORAS}^{-1}(\mathbf{F} - A\mathbf{U}^n) \tag{4.12}$$

yields iterates that are equivalent to that of the discretization of P.L. Lions' Algorithm (4.9)-(4.10), see [SCGT07b].

The third step is the introduction of a symmetric variant that allows for a comprehensive theoretical study. It seems at first glance that we should mimic what was done for the RAS algorithm and study the following symmetrized variant:

$$M_{OAS,1}^{-1} := \sum_{i=1}^{N} R_i^T B_i^{-1} R_i \,. \tag{4.13}$$

For reasons explained in Remark 9 which is coming later, we introduce another non standard variant of the ORAS preconditioner (4.11), the symmetrized ORAS (SORAS) algorithm:

$$M_{SORAS,1}^{-1} := \sum_{i=1}^{N} R_i^T D_i B_i^{-1} D_i R_i \,. \tag{4.14}$$

The missing step is the fourth one, namely to build an adaptive coarse space for a two-level SORAS method. It is done in the next section.

4.4 Mathematical framework & Theoretical analysis

In this section we will first state the mathematical context in order to reformulate and analyze the symmetrized ORAS (i.e SORAS) then introduce the two-level method with its analysis.

Given a Hilbert space V, a symmetric positive definite bilinear form $a : V \times V \longrightarrow \mathbb{R}$, and an element l in the dual space V', we consider the following variational problem on a domain $\Omega \subset \mathbb{R}^d$ for $d \in \mathbb{N}$

$$\text{Find } u \in V \text{ such that : } a_\Omega(u, v) = \ell(v) \quad , \forall v \in V' \quad , \tag{4.15}$$

a_Ω is defined in terms of an integral over any open set $\omega \subset \Omega$. Typical examples are the Darcy

equation (**K** is a diffusion tensor)

$$a_\omega(u,v) := \int_\omega \mathbf{K}\,\nabla u \cdot \nabla v\, dx\,,$$

or the elasticity system ($\underline{\underline{\underline{\underline{\boldsymbol{C}}}}}$ is the fourth-order stiffness tensor and $\underline{\underline{\boldsymbol{\varepsilon}}}(\underline{\boldsymbol{u}})$ is the strain tensor of a displacement field $\underline{\boldsymbol{u}}$):

$$a_\omega(\underline{\boldsymbol{u}},\,\underline{\boldsymbol{v}}) := \int_\omega \boldsymbol{C} : \underline{\underline{\boldsymbol{\varepsilon}}}(\underline{\boldsymbol{u}}) : \underline{\underline{\boldsymbol{\varepsilon}}}(\underline{\boldsymbol{v}})\, dx\,.$$

In order to use a finite element method, we consider a discretization of the variational problem (4.15) based on the mesh $\mathcal{T}_h$ of Ω

$$\Omega := \bigcup_{\tau\in\mathcal{T}_h} \tau$$

Let $V_h \subset V$ be the discrete variational space of shape functions, V_h can be a space of vector function in the case where the variational formulation derived from a system of PDEs. The discrete form of the variational problem (4.15) then reads

$$\text{Find } \mathrm{u}_h \in V_h \text{ such that : } a_\Omega(\mathrm{u}_h, v_h) = \ell(v_h) \quad ,\forall v_h \in V_h' \quad . \tag{4.16}$$

Let $\mathcal{N}$ denote the set of degrees of freedom $\#\mathcal{N} := dim(V_h)$ and $(\phi_k)_{k\in\mathcal{N}}$ be the finite elements basis on mesh $\mathcal{T}_h$

The finite element methods then leads to the following linear system

$$A\mathbf{U} = \mathbf{F} \tag{4.17}$$

where $(A_{ij})_{1\leqslant i,j\leqslant \#\mathcal{N}} = a_\Omega(\phi_j,\phi_i)$ and $(\mathbf{F}_i)_{i=1}^{\#\mathcal{N}} = \ell_\Omega(\phi_i)$.

Domain Ω is decomposed into N overlapping subdomains $(\Omega_i)_{1\leq i\leq N}$ so that all subdomains are a union of cells of the mesh $\mathcal{T}_h$. This decomposition induces a natural decomposition of the set of indices $\mathcal{N}$ into N subsets of indices $(\mathcal{N}_i)_{1\leq i\leq N}$:

$$\mathcal{N}_i := \{k \in \mathcal{N} \mid meas(\text{supp}\,\phi_k \cap \Omega_i) > 0\}\,,\ 1 \leq i \leq N. \tag{4.18}$$

For all $1 \leq i \leq N$, let R_i be the restriction matrix from $\mathbb{R}^{\#\mathcal{N}}$ to the subset $\mathbb{R}^{\#\mathcal{N}_i}$ and D_i be a diagonal matrix of size $\#\mathcal{N}_i \times \#\mathcal{N}_i$, so that we have a partition of unity at the algebraic level,

$$\sum_{i=1}^{N} R_i^T D_i R_i = I_d\,, \tag{4.19}$$

where $I_d \in \mathbb{R}^{\#\mathcal{N}\times\#\mathcal{N}}$ is the identity matrix.

For all subdomains $1 \leq i \leq N$, let B_i be a SPD matrix of size $\#\mathcal{N}_i \times \#\mathcal{N}_i$, which comes typically from the discretization of boundary value local problems using optimized transmission conditions.

We also define for all subdomains $1 \leq j \leq N$, $\widetilde{A}^j$, the $\#\mathcal{N}_j \times \#\mathcal{N}_j$ matrix defined by

$$\mathbf{V}_j^T \widetilde{A}^j \mathbf{U}_j := a_{\Omega_j} \left(\sum_{l \in \mathcal{N}_j} \mathbf{U}_{jl} \phi_l, \sum_{k \in \mathcal{N}_j} \mathbf{V}_{jk} \phi_k \right) , \quad \mathbf{U}_j, \mathbf{V}_j \in \mathbb{R}^{\mathcal{N}_j} . \tag{4.20}$$

When the bilinear form a results from the variational solution of a Laplace problem, the previous matrix corresponds to the discretization of local Neumann boundary value problems. For this reason we will call it "Neumann" matrix even in a more general setting.

We also make use of two numbers k_0 and k_1 related to the domain decomposition. Let

$$k_0 := \max_{1 \leq i \leq N} \# \left\{ j \mid R_j A R_i^T \neq 0 \right\} \tag{4.21}$$

be the maximum multiplicity of the interaction between subdomains plus one. Let k_1 be the maximal multiplicity of subdomains intersection, i.e. the largest integer m such that there exists m different subdomains whose intersection has a non zero measure.

Throughout the rest of this section we will show how the very abstract theory of the "Fictitious Space Lemma" [Nep91; Nep92] can be applied to better formalize the Optimized Schwarz Method which will allow to bound the condition number with one level preconditioner (ie. $\mathrm{M}_{\mathrm{SORAS},1}^{-1}\mathrm{A}$) then with a two-level preconditioner (ie. $\mathrm{M}_{\mathrm{SORAS},2}^{-1}\mathrm{A}$) and thus to confirm the robustness of the new coarse space.

4.4.1 One-Level SORAS

Here we are interested in the field of One-Level Optimized Schwarz. with the symmetric version as proposed in (4.14).

$$\mathrm{M}_{\mathrm{SORAS},1}^{-1} := \sum_{i=1}^{N} R_i^T D_i \mathrm{B}_i^{-1} D_i R_i,$$

where B_i is the local matrix problem with optimized interface condition.

In "Fictituous Space Lemma" [Nep91; Nep92] a certain number of abstract ingredients are needed. These ingredients can be easily identified in the case of Optimized Schwarz Method. We will give here the most important tools for the theoretical analysis.

Definition 15 (One-Level SORAS in fictitious space lemma)
Let H and H_D be two Hilbert spaces, each one is endowed with a scalar product

- *$H = \mathbb{R}^{\#\mathcal{N}}$ a Hilbert spaces endowed with the standard Euclidian scalar product*

$$\begin{aligned} a : & H \times H \longrightarrow \mathbb{R} \\ & (U, V) \longrightarrow a(U, V) = V^T A U, \end{aligned}$$

A is the matrix of the problem we want to solve. and

- $H_D = \prod_{i=1}^{N} \mathbb{R}^{\#\mathcal{N}_i}$ *is endowed with standard scalar Euclidian product. For* $\mathcal{U} = (\mathbf{U}_i)_{1\leqslant i\leqslant N}$, $\mathcal{V} = (\mathbf{V}_i)_{1\leqslant i\leqslant N}$ *with* $\mathbf{U}_i, \mathbf{V}_i \in \mathbb{R}^{\#\mathcal{N}_i}$, *the bilinear form* b *is defined by*

$$\begin{aligned} b : H_D \times H_D &\longrightarrow \mathbb{R} \\ (\mathcal{U}, \mathcal{V}) &\longmapsto b(\mathcal{U}, \mathcal{V}) := \sum_{i=1}^{N} \mathbf{V}_i^T B_i \mathbf{U}_i = \mathcal{V}^T \mathcal{B}\mathcal{U}, \end{aligned} \tag{4.22}$$

  Let B *denote the block-diagonal operator such that for all* $\mathcal{U}, \mathcal{V} \in H_D$, *we have:*

$$(B\mathcal{U}, \mathcal{V}) = b(\mathcal{U}, \mathcal{V})$$

- *Which makes possible to define the new operator* $\mathcal{B}$

$$\begin{aligned} \mathcal{B} : H_D &\longrightarrow H_D \\ \mathcal{U} &\longrightarrow \mathcal{B}\mathcal{U} = (B_i \mathbf{U}_i)_{1\leqslant i\leqslant N}. \end{aligned}$$

- *We can write its inverse as*

$$\begin{aligned} \mathcal{B}^{-1} : H_D &\longrightarrow H_D \\ \mathcal{U} &\longrightarrow \mathcal{B}^{-1}\mathcal{U} = \left(B_i^{-1} \mathbf{U}_i\right)_{1\leqslant i\leqslant N} \end{aligned}$$

- *For any* $\mathcal{U} = (\mathbf{U}_i)_{1\leqslant i\leqslant N}$ *the linear operator* $\mathcal{R}_{SORAS,1}$ *is defined as*

$$\begin{aligned} \mathcal{R}_{SORAS,1} : H_D &\longrightarrow H \\ \mathcal{U} &\longrightarrow \mathcal{R}_{SORAS,1}\mathcal{U} = \sum_{i=1}^{N} R_i^T D_i \mathbf{U}_i. \end{aligned}$$

Lemma 16
The One-Level preconditioner $M^{-1}_{SORAS,1}$ *can be rewritten as*

$$M^{-1}_{SORAS,1} = \sum_{i=1}^{N} R_i^T D_i B_i^{-1} D_i R_i = \mathcal{R}_{SORAS,1} \mathcal{B}^{-1} \mathcal{R}^*_{SORAS,1} \tag{4.23}$$

where $\mathcal{R}^*_{SORAS,1} : H \longrightarrow H_D$ *is the adjoint operator of* $\mathcal{R}_{SORAS,1}$ *with respect to the standard Euclidean scalar products.*

Proof. First of all, note that by definition $\mathcal{R}^*_{\text{SORAS},1}$ can be written as

$$< \mathcal{R}^*_{\text{SORAS},1}(\mathbf{U}), \mathcal{V} >_2 = < U, \mathcal{R}_{\text{SORAS},1}(\mathcal{V}) >_2 \text{ For all } \mathbf{U} \in H, \mathcal{V} := (\mathbf{V}_i)_{1\leqslant i\leqslant N} \in H_D,$$

in other words

$$\sum_{i=1}^{N} \mathbf{V}_i^T \left(\mathcal{R}^*_{\text{SORAS},1}(\mathbf{U})\right)_i = \mathcal{R}_{\text{SORAS},1}(\mathcal{V})^T \mathbf{U},$$

that is

$$\sum_{i=1}^{N} \mathbf{V}_i^T \left(\mathcal{R}^*_{\text{SORAS},1}(\mathbf{U})\right)_i = \left(\sum_{i=1}^{N} R_i^T D_i \mathbf{V}_i\right)^T \mathbf{U} = \sum_{i=1}^{N} \mathbf{V}_i^T D_i R_i \mathbf{U}.$$

Since this equality is valid for arbitrary $\mathbf{V}_i$, we have the identification:

$$\mathcal{R}^*_{\text{SORAS},1}(\mathbf{U}) = (D_i R_i U)_{1 \leqslant i \leqslant N}\,. \tag{4.24}$$

which allows us to write the symmetric Optimized Schwarz Method (4.14) in the form of (4.23). ■

The explanation of the application of the preconditionner in term of these operators is the following

- According to (4.24), the rightmost operator $\mathcal{R}^*_{\text{SORAS},1}$ decomposes a global vector in H into local components in H_D.
- The middle operator B^{-1} corresponds to solving N local problems.
- $\mathcal{R}_{\text{SORAS},1}$ interpolates the modified local components into global vector in H.

Now, we recall from [DJN15] the Fictitious Space Lemma [Nep91; Nep92], which will be our guiding environment.

Lemma 17 (Fictitious Space Lemma)
Let H and H_D be two Hilbert spaces, with the scalar products denoted by $(\cdot,\cdot)$ and $(\cdot,\cdot)_D$. Let the symmetric positive bilinear forms $a : H \times H \to \mathbb{R}$ and $b : H_D \times H_D \to \mathbb{R}$, generated by the s.p.d. operators $A : H \to H$ and $B : H_D \to H_D$, respectively (i.e. $(A\mathbf{U}, \mathbf{V}) = a(\mathbf{U}, \mathbf{V})$ for all $\mathbf{U}, \mathbf{V} \in H$ and $(B\mathcal{U}, \mathcal{V})_D = b(\mathcal{U}, \mathcal{V})$ for all $\mathcal{U}, \mathcal{V} \in H_D$).
Suppose that there exists a linear operator $\mathcal{R} : H_D \to H$, such that

- *$\mathcal{R}$ is onto.*
- *there exists a positive constant c_R such that*

$$a(\mathcal{R}\mathcal{U}, \mathcal{R}\mathcal{U}) \leq c_R \cdot b(\mathcal{U},\mathcal{U}) \;\; \forall \mathcal{U} \in H_D\,. \tag{4.25}$$

- *there exists a positive constant c_T such that for all $\mathbf{U} \in H$ there exists $\mathcal{U} \in H_D$ with $\mathcal{R}\mathcal{U} = \mathbf{U}$ and*

$$c_T \cdot b(\mathcal{U},\mathcal{U}) \leq a(\mathcal{R}\mathcal{U}, \mathcal{R}\mathcal{U}) = a(\mathbf{U}, \mathbf{U})\,. \tag{4.26}$$

We introduce the adjoint operator $\mathcal{R}^* : H \to H_D$ by $(\mathcal{R}\mathcal{U},\, \mathbf{U}) = (\mathcal{U},\, \mathcal{R}^*\mathbf{U})_D$ for all $\mathcal{U} \in H_D$ and $\mathbf{U} \in H$.
Then, we have the following spectral estimate

$$c_T \cdot a(\mathbf{U}, \mathbf{U}) \leq a\left(\mathcal{R}B^{-1}\mathcal{R}^* A\mathbf{U},\, \mathbf{U}\right) \leq c_R \cdot a(\mathbf{U}, \mathbf{U})\,, \;\; \forall \mathbf{U} \in H \tag{4.27}$$

which proves that the eigenvalues of operator $\mathcal{R}B^{-1}\mathcal{R}^*A$ are bounded from below by c_T and from above by c_R with sharp bounds for the spectrum of $\mathcal{R}B^{-1}\mathcal{R}^*A$ given by the best possible constants c_T and c_R in the above inequalities.

We can find the proof of the lemma 17 in the book [DJN15] page 165.

In our case $\mathcal{R} = \mathcal{R}_{\text{SORAS},1}$, we will verify in the following lemmas the assumptions of the fictituous space lemma 17.

Lemma 18 (surjectivity of $\mathcal{R}_{\text{SORAS},1}$)
The operator $\mathcal{R}_{SORAS,1}$ defined by

$$\begin{aligned}\mathcal{R}_{SORAS,1} :& H_D \longrightarrow H \\ & \mathcal{U} \longrightarrow \mathcal{R}_{SORAS,1}\mathcal{U} = \sum_{i=1}^{N} R_i^T D_i \mathbf{U}_i\end{aligned}$$

is onto.

Proof. The proof is straightforward, from the partition of unity

$$Id = \sum_{i=1}^{N} R_i^T D_i R_i$$

we have for all $\mathbf{U} \in H$,

$$\mathbf{U} = \sum_{i=1}^{N} R_i^T D_i R_i \mathbf{U}.$$

Which allows us to take $\mathcal{U} = ((R_i\mathbf{U})_{1\leqslant i\leqslant N})$. ■

Lemma 19 (Continuity of $\mathcal{R}_{\text{SORAS},1}$)
Let k_0 be defined as

$$k_0 := \max_{1\leqslant i\leqslant N} \#\{j \mid R_i A R_j^T \neq 0\},$$

and γ by

$$\gamma := \max_{1\leqslant i\leqslant N} \max_{\mathbf{U}_i\in\mathbb{R}^{\#\mathcal{N}_i}\setminus 0} \frac{\left(R_i^T D_i \mathbf{U}_i\right)^T A \left(R_i^T D_i \mathbf{U}_i\right)}{\mathbf{U}_i^T B_i \mathbf{U}_i}. \tag{4.28}$$

Then the linear operator $\mathcal{R}_{SORAS,1}$ is continuous with constant $k_0\gamma$, ie $\forall \mathcal{U} \in H_D$ we have the following continuity estimate

$$a\left(\mathcal{R}_{SORAS,1}\mathcal{U}, \mathcal{R}_{SORAS,1}\mathcal{U}\right) \leqslant k_0\gamma b(\mathcal{U},\mathcal{U}).$$

Proof. Let $\mathcal{U} = (\mathbf{U}_i)_{0\leqslant i\leqslant N} \in H_D$, then using the lemma 33 in Appendix B and the definition of γ in (4.28), we get

$$\begin{aligned} a\left(\mathcal{R}_{\text{SORAS},1}\mathcal{U}, \mathcal{R}_{\text{SORAS},1}\mathcal{U}\right) &= \left(\mathcal{R}_{\text{SORAS},1}\mathcal{U}\right)^T A \left(\mathcal{R}_{\text{SORAS},1}\mathcal{U}\right) \\ &= \left(\sum_{i=1}^{N} R_i^T D_i \mathbf{U}_i\right)^T A \left(\sum_{j=1}^{N} R_j^T D_j \mathbf{U}_j\right) \\ &\leqslant k_0 \sum_{i=1}^{N} \left(R_i^T D_i \mathbf{U}_i\right)^T A \left(R_i^T D_i \mathbf{U}_i\right) \\ &\leqslant k_0\gamma \sum_{i=0}^{N} \mathbf{U}_i^T \mathrm{B}_i \mathbf{U}_i = \underbrace{k_0\gamma}_{C_R} b\left(\mathcal{U},\mathcal{U}\right) \end{aligned}$$

■

Lemma 20 (stable decomposition with $\mathcal{R}_{\text{SORAS},1}$)
Let τ be defined as

$$\tau := \min_{1\leqslant i\leqslant N} \min_{\mathbf{U}_i\in\mathbb{R}^{\#\mathcal{N}_i}\setminus 0} \frac{\mathbf{U}_i^T \tilde{A}_i \mathbf{U}_i}{\mathbf{U}_i^T B_i \mathbf{U}_i} \tag{4.29}$$

let $c_T := \tau^{-1} k_1$.
Then, $\forall U \in H$ there exists $\mathcal{U} \in H_D$

$$c_T b(\mathcal{U},\mathcal{U}) \leqslant a(\mathcal{R}_{SORAS,1}\mathcal{U}, \mathcal{R}_{SORAS,1}\mathcal{U}).$$

then, the sequence $U := (R_i U)_{1\leqslant i\leqslant N}$ forms a stable decomposition of U with a constant $\tau^{-1}k_1$. that is:

- $$b(\mathcal{U},\mathcal{U}) \leq \tau^{-1} k_1\, a(\mathbf{U},\mathbf{U}).$$

Proof. Let $\mathbf{U} \in H$ and $\mathcal{U} := (R_i\mathbf{U})_{1\leqslant i\leqslant N} \in H_D$ so that $\mathbf{U} = \mathcal{R}_{\text{SORAS},1}\mathcal{U}$,
Then using (4.29) and Lemma 34 in Appendix B, we get the following

$$\begin{aligned} b(\mathcal{U},\mathcal{U}) &= \sum_{i=1}^{N} \mathbf{U}_i^T \mathrm{B}_i \mathbf{U}_i \\ &= \sum_{i=1}^{N} (R_i\mathbf{U})^T B_i (R_i\mathbf{U}) \\ &\leqslant \tau^{-1} \sum_{i=1}^{N} (R_i\mathbf{U})^T \widetilde{A}^i (R_i\mathbf{U}) \\ &\leqslant \tau^{-1} k_1 \mathbf{U}^T A\mathbf{U}. \end{aligned}$$

■

Now, to sum up, with the results of Lemma 18, 19 and 20, we have proved the following.

i $\mathcal{R}_{\text{SORAS},1}$ is onto.

ii Continuity: $\exists c_R$ such that $\forall \mathcal{U} \in H_D\ a(\mathcal{R}_{\text{SORAS},1}\mathcal{U}, \mathcal{R}_{\text{SORAS},1}\mathcal{U}) \leqslant c_R b(\mathcal{U},\mathcal{U})$.

iii Stable decomposition: $\exists c_T$ such that $\forall \mathbf{U} \in H$ there exists $\mathcal{U} \in H_D$ with $\mathbf{U} = \mathcal{R}_{\text{SORAS},1}\mathcal{U}$ $c_T b(\mathcal{U},\mathcal{U}) \leqslant a(\mathcal{R}_{\text{SORAS},1}\mathcal{U}, \mathcal{R}_{\text{SORAS},1}\mathcal{U})$.

which are the assumptions of the fictitious space lemma and therefore, their virification allows us to get the following theorem.

Theorem 21
The eigenvalues of the One-Level Optimized Schwarz preconditioned system satisfy the following estimate

$$\frac{\tau}{k_1} \leqslant \kappa\left(M_{SORAS,1}^{-1} A\right) \leqslant k_0\gamma,$$

with γ defined as in (4.28)*, and τ as in* (4.29)*.*

4.4.2 SORAS with GenEO-2

We now consider a two-level method based on enriching the one-level SORAS preconditioner (4.11) by introducing two generalized eigenvalue problems which allow us to control the spectrum of the preconditioned operator as written in Theorem 31.

Coarse Space for the lower bound

More precisely, we define the following generalized eigenvalue problem:

Definition 22 (Generalized Eigenvalue Problem for the lower bound)
For each subdomain $1 \leqslant j \leqslant N$, we introduce the generalized eigenvalue problem

$$\begin{aligned} &\text{Find } (\mathbf{V}_{jk}, \lambda_{jk}) \in \mathbb{R}^{\#\mathcal{N}_j} \setminus \{0\} \times \mathbb{R} \text{ such that} \\ &\widetilde{A}^j \mathbf{V}_{jk} = \lambda_{jk} B_j \mathbf{V}_{jk}. \end{aligned} \tag{4.30}$$

Let $\tau > 0$ be a user-defined threshold, we define $Z^{\tau}_{geneo} \subset \mathbb{R}^{\#\mathcal{N}}$ as the vector space spanned by the family of vectors $(R_j^T D_j \mathbf{V}_{jk})_{\lambda_{jk}<\tau, 1\leqslant j\leqslant N}$ corresponding to eigenvalues smaller than τ.

Let $\tilde{\pi}_j$ be the projection from $\mathbb{R}^{\#\mathcal{N}_j}$ on Span$\{\mathbf{V}_{jk} | \lambda_{jk} < \tau\}$ parallel to Span$\{\mathbf{V}_{jk} | \lambda_{jk} \geq \tau\}$. In the present case of the SORAS-2 method, Lemma 32 in Appendix B translates into:

Lemma 23 (Intermediate Lemma for GenEO-SORAS-2)
For all subdomains $1 \leqslant j \leqslant N$ and $\mathbf{U}_j \in \mathbb{R}^{\mathcal{N}_j}$, we have:

$$\tau \ ((I_d - \tilde{\pi}_j)\mathbf{U}_j)^T B_j (I_d - \tilde{\pi}_j)\mathbf{U}_j \leqslant \ \mathbf{U}_j^T \widetilde{A}^j \mathbf{U}_j \, . \tag{4.31}$$

Coarse space for the upper bound

We introduce the following generalized eigenvalue problem:

Definition 24 (Generalized Eigenvalue Problem for the upper bound)

$$\begin{aligned} &\text{Find } (\mathbf{U}_{ik}, \mu_{ik}) \in \mathbb{R}^{\#\mathcal{N}_i} \setminus \{0\} \times \mathbb{R} \text{ such that} \\ &D_i R_i A R_i^T D_i \mathbf{U}_{ik} = \mu_{ik} B_i \, \mathbf{U}_{ik}. \end{aligned} \tag{4.32}$$

Let $\gamma > 0$ be a user-defined threshold, we define $Z^{\gamma}_{geneo} \subset \mathbb{R}^{\#\mathcal{N}}$ as the vector space spanned by the family of vectors $(R_i^T D_i \mathbf{U}_{ik})_{\mu_{ik}>\gamma, 1\leqslant i\leqslant N}$ corresponding to eigenvalues larger than γ.

Now, let ξ_i denote the projection from $\mathbb{R}^{\mathcal{N}_i}$ on Span $\{\mathbf{U}_{ik} \,|\, \gamma > \mu_{ik}\}$ parallel to Span $\{\mathbf{U}_{ik} \,|\, \gamma \leqslant \mu_{ik}\}$. From these Definitions, Lemma 32 in Appendix B leads to:

Lemma 25
For all subdomain $1 \leqslant i \leqslant N$ and $\mathbf{U}_i \in \mathbb{R}^{\#\mathcal{N}_i}$, we have:

$$\left(R_i^T D_i (I_d - \xi_i)\mathbf{U}_i\right)^T A \, R_i^T D_i (I_d - \xi_i)\mathbf{U}_i) \leqslant \gamma \, \mathbf{U}_i^T B_i \mathbf{U}_i \, . \tag{4.33}$$

We are now ready to define the SORAS two level preconditioner

Definition 26 (Two level SORAS-GENEO-2 preconditioner)
Let P_0 denote the a-orthogonal projection on the SORAS-GENEO-2 coarse space

$$Z_{\text{GenEO-2}} := Z^{\tau}_{geneo} \bigoplus Z^{\gamma}_{geneo} ,$$

the two-level SORAS-GENEO-2 preconditioner is defined as follows, see [Man92]:

$$\boxed{M^{-1}_{SORAS,2} := P_0 A^{-1} + (I_d - P_0) \sum_{i=1}^{N} R_i^T D_i B_i^{-1} D_i R_i (I_d - P_0^T) .} \tag{4.34}$$

Let Z_0 be a matrix whose columns are a basis of $Z_{\text{GenEO-2}}$ and let denote its transpose by $R_0 := Z_0^T$. It is easily checked that

$$P_0 A^{-1} = R_0^T (R_0 A R_0^T)^{-1} R_0 .$$

This definition is reminiscent of the balancing domain decomposition preconditioner [Man92] introduced for Schur complement based methods. Note that the coarse space is now defined by two generalized eigenvalue problems instead of one in [SDHNPS14; SDHNR13] for ASM and FETI-BDD methods.

In the following we conduct an analysis of the Two-Level SORAS method, which will lead later to the Theorem 31. The proof of Theorem 31 is also based on the Fictitious Space [Nep91] recalled in the present chapter in lemma 17.

Definition 27 (Two-level SORAS in the Fictitious Space Lemma)
Two Hilbert spaces H and H_D, two other associated bilinear forms and induced scalar products as well as the $\mathcal{R}_{SORAS,2}$ operator between them are defined as follows.

- *Space $H := \mathbb{R}^{\#\mathcal{N}}$ endowed with the standard Euclidian scalar product. We consider another bilinear form a defined by :*

$$a : H \times H \to \mathbb{R}, \quad (\mathbf{U}, \mathbf{V}) \longmapsto a(\mathbf{U}, \mathbf{V}) := \mathbf{V}^T A \mathbf{U}. \tag{4.35}$$

  where A is the matrix of the problem we want to solve.

- *Space H_D is defined as the product space*

$$H_D := \mathbb{R}^{\#\mathcal{N}_0} \times \prod_{i=1}^{N} \mathbb{R}^{\#\mathcal{N}_i} \tag{4.36}$$

  is endowed with standard scalar Euclidian product. For $\mathcal{U} = (\mathbf{U}_i)_{1 \leqslant i \leqslant N}$, $\mathcal{V} = (\mathbf{V}_i)_{1 \leqslant i \leqslant N}$ with $\mathbf{U}_i, \mathbf{V}_i \in \mathbb{R}^{\#\mathcal{N}_i}$, the bilinear form b is defined by

$$\begin{aligned} b : H_D \times H_D &\longrightarrow \mathbb{R} \\ (\mathcal{U}, \mathcal{V}) &\longmapsto b(\mathcal{U}, \mathcal{V}) := (R_0^T \mathbf{V}_0)^T A \, (R_0^T \mathbf{U}_0) + \sum_{i=1}^{N} \mathbf{V}_i^T B_i \mathbf{U}_i. \end{aligned} \tag{4.37}$$

Let B denote the block-diagonal operator such that for all $\mathcal{U}, \mathcal{V} \in H$, we have:

$$(B\mathcal{U}, \mathcal{V}) = b(\mathcal{U}, \mathcal{V}).$$

– *For any $\mathcal{U} = (\mathbf{U}_i)_{0 \leqslant i \leqslant N}$ the linear operator $\mathcal{R}_{SORAS,2}$ is defined as*

$$\mathcal{R}_{SORAS,2} : H_D \longrightarrow H,\ \mathcal{R}_{SORAS,2}(\mathcal{U}) := R_0^T\, \mathbf{U}_0 + \sum_{i=1}^{N} (I_d - P_0)\, R_i^T D_i\, \mathbf{U}_i. \tag{4.38}$$

It can easily be checked that

$$M_{SORAS,2}^{-1} = \mathcal{R}_{\text{SORAS},2} B^{-1} \mathcal{R}_{\text{SORAS},2}^{*},$$

where $\mathcal{R}_{\text{SORAS},2}^{*} : H \longrightarrow H_D$ is the adjoint operator of $\mathcal{R}_{\text{SORAS},2}$ with respect to the standard Euclidean scalar product.

We now check the assumptions of the Fictitious Space Lemma.

Lemma 28 (Surjectivity of $\mathcal{R}_{\text{SORAS},2}$)
The operator $\mathcal{R}_{SORAS,1}$ defined by

$$\begin{aligned} \mathcal{R}_{SORAS,2} :& H_D \longrightarrow H \\ & \mathcal{U} \longrightarrow \mathcal{R}_{SORAS,2}\mathcal{U} = R_0^T U_0 + \sum_{i=1}^{N} (I_d - P_0)\, R_i^T D_i \mathbf{U}_i, \end{aligned}$$

is onto.

Proof. For all $\mathbf{U} \in H$, we have:

$$\mathbf{U} = P_0\, \mathbf{U} + (I_d - P_0)\, \mathbf{U} = P_0\, \mathbf{U} + \sum_{i=1}^{N} (I_d - P_0)\, R_i^T D_i R_i \mathbf{U}\,.$$

Since $P_0\, \mathbf{U} \in \text{Span}(R_0^T)$, there exists $\mathbf{U}_0 \in R^{\#\mathcal{N}_0}$ such that $P_0\, \mathbf{U} = R_0^T\, \mathbf{U}_0$. Thus, we have

$$\mathbf{U} = R_0^T\, \mathbf{U}_0 + \sum_{i=1}^{N} (I_d - P_0)\, R_i^T D_i (R_i \mathbf{U})\,,$$

or, in other words

$$\mathcal{R}_{\text{SORAS},2}(\mathbf{U}_0, (R_i \mathbf{U})_{1 \leqslant i \leqslant N}) = \mathbf{U}\,,$$

which proves the surjectivity. ■

We now prove

Lemma 29 (Continuity of $\mathcal{R}_{\text{SORAS},2}$)
Let $\mathcal{U} = (\mathbf{U}_i)_{0 \leqslant i \leqslant N} \in H$. We have the following continuity estimate

$$a(\mathcal{R}_{SORAS,2}(\mathcal{U}), \mathcal{R}_{SORAS,2}(\mathcal{U})) \leqslant \max(1, k_0\, \gamma)\, b(\mathcal{U}, \mathcal{U})\,.$$

Proof. Since P_0 and $I_d - P_0$ are a-orthogonal projections, we have by a-orthogonality:

$$\begin{aligned} a(\mathcal{R}_{\text{SORAS},2}(\mathcal{U}), \mathcal{R}_{\text{SORAS},2}(\mathcal{U})) &= a\left(P_0\, R_0^T\, \mathbf{U}_0, P_0\, R_0^T\, \mathbf{U}_0\right) \\ &\quad + a\left((I_d - P_0) \sum_{i=1}^{N} R_i^T \mathrm{D}_i\, \mathbf{U}_i, (I_d - P_0) \sum_{i=1}^{N} R_i^T \mathrm{D}_i\, \mathbf{U}_i\right) \end{aligned}$$

Since P_0 is the a-orthogonal projection on $Z_{\text{GenEO-2}}$ and that

$$\sum_{i=1}^{N} R_i^T D_i \xi_i \mathbf{U}_i \in Z_{geneo}^{\gamma} \subset Z_{\text{GenEO-2}}\,,$$

we have

$$(I_d - P_0) \sum_{i=1}^{N} R_i^T D_i \xi_i \mathbf{U}_i = 0\,,$$

and thus

$$\begin{aligned} &a\left((I_d - P_0) \sum_{i=1}^{N} R_i^T \mathrm{D}_i\, \mathbf{U}_i, (I_d - P_0) \sum_{i=1}^{N} R_i^T \mathrm{D}_i\, \mathbf{U}_i\right) \\ &\quad = a\left((I_d - P_0) \sum_{i=1}^{N} R_i^T \mathrm{D}_i\, (I_d - \xi_i)\mathbf{U}_i, (I_d - P_0) \sum_{i=1}^{N} R_i^T \mathrm{D}_i\, (I_d - \xi_i)\mathbf{U}_i\right)\,. \end{aligned}$$

Finally, using k_0 defined as in Lemma 33 in appendix B, we have

$$\begin{aligned} a(\mathcal{R}_{\text{SORAS},2}(\mathcal{U}), \mathcal{R}_{\text{SORAS},2}(\mathcal{U})) &\leqslant a\left(\, R_0^T\, \mathbf{U}_0,\, R_0^T\, \mathbf{U}_0\right) \\ &\quad + a\left(\sum_{i=1}^{N} R_i^T \mathrm{D}_i\, (I_d - \xi_i)\mathbf{U}_i, \sum_{i=1}^{N} R_i^T \mathrm{D}_i\, (I_d - \xi_i)\mathbf{U}_i\right) \\ &\leq a\left(\, R_0^T\, \mathbf{U}_0,\, R_0^T\, \mathbf{U}_0\right) \\ &\quad + k_0 \sum_{i=1}^{N} a\left(R_i^T \mathrm{D}_i\, (I_d - \xi_i)\mathbf{U}_i, R_i^T \mathrm{D}_i\, (I_d - \xi_i)\mathbf{U}_i\right)\,. \end{aligned}$$

Then, using estimate (4.33), we have:

$$\begin{aligned} a(\mathcal{R}_{\text{SORAS},2}(\mathcal{U}), \mathcal{R}_{\text{SORAS},2}(\mathcal{U})) &\leq a(\, R_0^T\, \mathbf{U}_0,\, R_0^T\, \mathbf{U}_0) + k_0\, \gamma \sum_{i=1}^{N} (B_i\, \mathbf{U}_i,\, \mathbf{U}_i) \\ &\leq \max(1, k_0\, \gamma)\, b(\mathcal{U}, \mathcal{U})\,. \end{aligned}$$

which concludes the estimate of the continuity of $\mathcal{R}_{\text{SORAS},2}$. ■

Lemma 30 (Stable decomposition with $\mathcal{R}_{\text{SORAS},2}$)
Let $\mathbf{U}$ *be a vector in* H*. We define:*

$$\mathbf{U}_j := (I_d - \widetilde{\pi}_j)\, R_j\, \mathbf{U}$$

and $\mathbf{U}_0 \in \mathbb{R}^{\#\mathcal{N}_0}$ *such that:*

$$R_0^T\, \mathbf{U}_0 = P_0\, \mathbf{U}\,.$$

We define $\mathcal{U} := (\mathbf{U}_i)_{0 \leqslant i \leqslant N}$.
Then, the stable decomposition property is verified with a constant $(1 + k_1\,\tau^{-1})^{-1}$*, since we have:*

- $$\mathcal{R}_{SORAS\,,2}(\mathcal{U}) = \mathbf{U}\,,$$
- $$\frac{1}{(1 + k_1\,\tau^{-1})} b(\mathcal{U},\mathcal{U}) \leqslant\ a(\mathbf{U},\mathbf{U}).$$

Proof. We first check that we have indeed a decomposition $\mathcal{R}_{\text{SORAS}\,,2}(\mathcal{U}) = \mathbf{U}$. Note that for all $1 \leqslant j \leqslant N$ we have

$$\mathrm{R}_j^T\, D_j\, \widetilde{\pi}_j\, \mathrm{R}_j\, \mathbf{U} \in Z^{\tau}_{geneo} \subset Z_{\text{GenEO-2}} \Rightarrow (I_d - P_0)\mathrm{R}_j^T\, D_j\, \widetilde{\pi}_j\, \mathrm{R}_j\, \mathbf{U} = 0\,.$$

We have:

$$\begin{aligned}
\mathbf{U} &= P_0\mathbf{U} + (I_d - P_0)\mathbf{U} = P_0\mathbf{U} + (I_d - P_0)\sum_{j=1}^{N} \mathrm{R}_j^T \mathrm{D}_j\, \mathrm{R}_j\, \mathbf{U} \\
&= P_0 R_0^T\, \mathbf{U}_0 + (I_d - P_0)\sum_{j=1}^{N} \mathrm{R}_j^T D_j\, (I_d - \widetilde{\pi}_j)\, \mathrm{R}_j\, \mathbf{U} = \mathcal{R}_{\text{SORAS}\,,2}(\mathcal{U})\,.
\end{aligned}$$

The last thing to do is to check the stability of this decomposition. Using (4.31) then Lemma 34 in Appendix B , we have

$$\begin{aligned}
b(\mathcal{U},\mathcal{U}) &= a(R_0^T\mathbf{U}_0, R_0^T\mathbf{U}_0) \\
&\quad + \sum_{j=1}^{N} \left((I_d - \widetilde{\pi}_j)\, \mathrm{R}_j\, \mathbf{U}\right)^T B_j\, \left((I_d - \widetilde{\pi}_j)\, \mathrm{R}_j\, \mathbf{U}\right) \\
&\leqslant a(P_0\mathbf{U}, P_0\mathbf{U}) + \tau^{-1} \sum_{j=1}^{N} (\mathrm{R}_j\mathbf{U})^T \widetilde{A}^j (\mathrm{R}_j\mathbf{U}) \\
&\leq a(\mathbf{U},\mathbf{U}) + k_1\,\tau^{-1} a(\mathbf{U},\mathbf{U}) \leqslant (1 + k_1\,\tau^{-1})\, a(\mathbf{U},\mathbf{U}).
\end{aligned}$$

■

The assumptions of the Fictitious Space Lemma are verified and thus we have just proved the following

Theorem 31 (Spectral estimate for the two level SORAS-GenEO-2)
Let γ *be a chosen threshold in Def 24,* τ *be a chosen threshold in Def* (22) *of the GenEO-2 coarse space and the two-level SORAS-GenEO-2 preconditioner defined by* (4.34)*. Then, the eigenvalues of the two-level SORAS-GenEO-2 preconditioned system satisfy the following estimate*

$$\boxed{\frac{1}{1 + \frac{k_1}{\tau}} \leqslant \lambda(M^{-1}_{SORAS,2}\, A) \leqslant \max(1, k_0\,\gamma).}$$

We have the

Remark 9
An analysis of a two-level version of the preconditioner M_{OAS}^{-1} (4.13) following the same path yields the following two generalized eigenvalue problems:

$$\begin{array}{c}\text{Find } (\mathbf{U}_{jk}, \mu_{jk}) \in \mathbb{R}^{\#\mathcal{N}_i} \setminus \{0\} \times \mathbb{R} \text{ such that} \\ A^i \mathbf{U}_{ik} = \mu_{ik} B_i \mathbf{U}_{ik} \ ,\end{array}$$

and

$$\begin{array}{c}\text{Find } (\mathbf{V}_{jk}, \lambda_{jk}) \in \mathbb{R}^{\#\mathcal{N}_i} \setminus \{0\} \times \mathbb{R} \text{ such that} \\ \widetilde{A}^i \mathbf{V}_{ik} = \lambda_{ik} D_i B_i D_i \mathbf{V}_{ik} \ .\end{array} .$$

In the general case for $1 \leqslant i \leqslant N$, matrices D_i may have zero entries for boundary degrees of freedom since they are related to a partition of unity. Moreover very often matrices B_i and A_i differ only by the interface conditions that is for entries corresponding to boundary degrees of freedom. Therefore, matrix $D_i B_i \mathrm{D}_i$ on the right hand side of the last generalized eigenvalue problem is not impacted by the choice of the interface conditions of the one level optimized Schwarz method. This cannot lead to efficient adaptive coarse spaces.

4.5 Saddle point problems

Many applications in science and engineering require solving large linear algebraic systems in saddle point form; see [BGL05] for an extensive survey. Although our theory does not apply in a straightforward manner to saddle point problems, we use it for these difficult problems for which it is not always possible to preserve both symmetry and positivity of the problem, see [LP08]. Note that generalized eigenvalue problems (4.32) and (4.30) still make sense if A is the matrix of a saddle point problem and local matrices A_i, B_i and $\widetilde{A}_i$, $1 \leq i \leq N$, are based on a partition of unity and on variational formulations.

We start by the global problem defined via variational formulation see for instance § 4.6.1 for the systems of almost incompressible elasticity and of Stokes. As in section 4.4, these formulations are written in terms of integrals of differential quantities (gradient, divergence, ...) over some domain $\Omega \subset \mathbb{R}^d$ for $d \in \mathbb{N}$:

Find $(\boldsymbol{u}, p) \in V \times \Lambda$ such that:

$$\begin{array}{rcl} a_\Omega(\boldsymbol{u},\, \boldsymbol{v}) + b_\Omega(\boldsymbol{v},\, p) & = & l_1(\boldsymbol{v})\,, \ \forall \boldsymbol{v} \in V\,, \\ b_\Omega(\boldsymbol{u},\, q) - c_\Omega(p,\, q) & = & l_2(q)\,, \ \forall q \in \Lambda\,, \end{array}$$

where V and Λ are Hilbert spaces of functions from Ω with real values, a_Ω, b_Ω and c_Ω are bilinear forms, a_Ω being symmetric. Discretization by a finite element method yields a saddle point system of the form:

$$A := \begin{bmatrix} H & B^T \\ B & -C \end{bmatrix} \begin{bmatrix} \boldsymbol{u} \\ p \end{bmatrix} = \begin{bmatrix} \boldsymbol{f} \\ g \end{bmatrix}\,, \tag{4.39}$$

where $H = H^T$ is positive definite, $C = C^T$ is positive semidefinite. The set of degrees of freedom is decomposed into subsets $(\mathcal{N}_i)_{1 \leq i \leq N}$. The matrices involved in the partition of unity (4.19)

have a block diagonal form

$$D_i := \begin{bmatrix} D_i^u & 0 \\ 0 & D_i^p \end{bmatrix} \text{ and } R_i := \begin{bmatrix} R_i^u & 0 \\ 0 & R_i^p \end{bmatrix} .$$

The local "Dirichlet" matrices have the following block form:

$$A_i := R_i A R_i^T = \begin{bmatrix} H_i & B_i^T \\ B_i & -C_i \end{bmatrix},$$

where

$$H_i := R_i^u H R_i^{u\,T} , \;\; C_i := R_i^p C R_i^{p\,T} \text{ and } B_i := R_i^p B R_i^{u\,T} .$$

The local "Neumann" problems arise from the variational formulation restricted the finite element space of a subdomain as in (4.20). We use the following block notation

$$\widetilde{A}_i := \begin{bmatrix} \widetilde{H}_i & \widetilde{B}_i^T \\ \widetilde{B}_i & -\widetilde{C}_i \end{bmatrix} .$$

For each subdomain $1 \leq i \leq N$, the "Robin" matrix is

$$B_i = \widetilde{A}_i + Z_i$$

where $Z_i = Z_i^T$ is positive semidefinite and is such that matrix B_i is symmetric positive definite. For sake of simplicity the "Robin" boundary condition will only apply to the $\boldsymbol{u}$ term, that is:

$$Z_i = \begin{bmatrix} Z_i^u & 0 \\ 0 & 0 \end{bmatrix} .$$

4.5.1 GenEO eigenvalue problem for saddle point problems

Eigenvalue problem for saddle point problem has been considered by various authors, see [BS06] and references therein. We cannot use directly their results since we consider generalized eigenvalue problems where both left and right matrices have saddle point structures. In order to prove that the GenEO eigenvalues are real and non negative, we need the following assumption:

Assumption 1

$$(\widetilde{H}_i \boldsymbol{u},\, \boldsymbol{u}) + (Z_i^u \boldsymbol{u},\, \boldsymbol{u}) + (\widetilde{C}_i p, p) = 0 \Rightarrow \boldsymbol{u} = 0 \textit{ and } p = 0. \tag{4.40}$$

Consider the generalized eigenvalue problem that controls the lower bound of the spectrum of the preconditioned system:

$$\begin{bmatrix} \widetilde{H}_i & \widetilde{B}_i^T \\ \widetilde{B}_i & -\widetilde{C}_i \end{bmatrix} \begin{bmatrix} \boldsymbol{u} \\ p \end{bmatrix} = \lambda \begin{bmatrix} \widetilde{H}_i + Z_i^u & \widetilde{B}_i^T \\ \widetilde{B}_i & -\widetilde{C}_i \end{bmatrix} \begin{bmatrix} \boldsymbol{u} \\ p \end{bmatrix} . \tag{4.41}$$

We take the scalar product of (4.41) with $\begin{bmatrix} \boldsymbol{u} & -p \end{bmatrix}^T$. The cross product terms $(\widetilde{B}_i^T \boldsymbol{u},\, p)$ cancel and we get:

$$(\widetilde{H}_i \boldsymbol{u},\, \boldsymbol{u}) + (\widetilde{C}_i p,\, p) = \lambda \left[(\widetilde{H}_i \boldsymbol{u},\, \boldsymbol{u}) + (Z_i^u \boldsymbol{u},\, \boldsymbol{u}) + (\widetilde{C}_i p, p) \right] . \tag{4.42}$$

All terms above are non negative. From Assumption 1, the right term cannot be zero. Therefore, $\lambda \in [0, 1]$.

Consider now the eigenvalue problem that controls the upper bound of the spectrum of the preconditioned system:

$$\begin{bmatrix} D_i^u H_i D_i^u & D_i^u B_i^T D_i^p \\ D_i^p B_i D_i^u & -D_i^p C_i \ D_i^p \end{bmatrix} \begin{bmatrix} \boldsymbol{u} \\ p \end{bmatrix} = \mu \begin{bmatrix} \widetilde{H}_i + Z_i^u & \widetilde{B}_i^T \\ \widetilde{B}_i & -\widetilde{C}_i \end{bmatrix} \begin{bmatrix} \boldsymbol{u} \\ p \end{bmatrix}. \tag{4.43}$$

We take the scalar product of (4.43) with $\begin{bmatrix} \boldsymbol{u} & -p \end{bmatrix}^T$ and we get:

$$(H_i D_i^u \boldsymbol{u},\, D_i^u \boldsymbol{u}) + (C_i D_i^p p,\, D_i^p p) = \mu \left[(\widetilde{H}_i \boldsymbol{u},\, \boldsymbol{u}) + (Z_i^u \boldsymbol{u},\, \boldsymbol{u}) + (\widetilde{C}_i p, p) \right] \tag{4.44}$$

All terms above are non negative. From Assumption 1, the right term cannot be zero. Therefore, $\mu \geq 0$.

4.6 Application to the systems of Stokes and of Nearly Incompressible elasticity

Mixed finite elements are often used to solve incompressible Stokes and nearly incompressible elasticity problems. Continuous pressures have been used in many mixed finite elements. However, most domain decomposition methods require that the pressure be discontinuous when they are used to solve the indefinite linear systems arising from such mixed finite element discretizations. Several domain decomposition algorithms allow one to use continuous pressures, see [TL15] and references therein. To our knowledge, our method is the first one to exhibit scalability for a highly heterogeneous nearly incompressible elasticity problems with continuous pressures.

4.6.1 Variational formulations

Consider a deformable solid which occupies the space domain Ω, we denote by $\partial\Omega$ the boundary of the Ω which is assumed to be a piecewise C^1, we denote by $\underline{\mathrm{u}}$ the displacement.
We look for the displacement $\underline{\mathrm{u}} : \Omega \longrightarrow \mathbb{R}^d$ with the following strong formulation

$$\begin{cases} -\underline{\nabla} \cdot (\underline{\underline{\sigma}}_S(\underline{\mathrm{x}})) & = \underline{f}(\underline{\mathrm{x}}), \quad \text{in } \Omega \\ \underline{\underline{\sigma}}_S(\underline{\mathrm{x}}) \cdot \underline{n} & = \underline{g}(\underline{\mathrm{x}}), \quad \text{on } \Gamma^N \\ \underline{\mathrm{u}}(\underline{\mathrm{x}}) & = 0 \quad \in \Gamma^D. \end{cases} \tag{4.45}$$

In the framework of linear Elasticity, we consider the following constitutive law

$$\underline{\underline{\sigma}}_S = \underline{\underline{\underline{A}}} : \underline{\underline{\varepsilon}}(\underline{\mathrm{u}}), \tag{4.46}$$

where $\underline{\underline{\varepsilon}}$ is the linearized strain tensor which is defined by

$$\underline{\underline{\varepsilon}}(\underline{\mathrm{u}}) = \frac{1}{2} \left(\underline{\underline{\nabla}}\, \underline{\mathrm{u}} + (\underline{\underline{\nabla}}\, \underline{\mathrm{u}})^T \right) \text{ with } \varepsilon_{i,j} = \frac{1}{2} \left(\frac{\partial \mathrm{u}_i}{\partial x_j} + \frac{\partial \mathrm{u}_j}{\partial x_i} \right), \tag{4.47}$$

and $\underline{\underline{\underline{A}}}$ is symmetric, in our case we consider a Saint-Venant Kirchhoff law

$$\underline{\underline{\sigma}}_S = \lambda \operatorname{tr}(\underline{\underline{\varepsilon}}) I_d + 2\mu \underline{\underline{\varepsilon}}. \tag{4.48}$$

λ and μ are the Lamé parameters, they can be expressed in terms of Young's modulus E and Poisson ratio ν

$$\lambda = \frac{E\nu}{(1+\nu)(1-2\nu)} \text{ and } \mu = \frac{E}{2(1+\nu)}.$$

Variational formulation : We consider an admissible function $\underline{v} \in \mathcal{V}$ where the variational space of admissible displacement is defined by $\mathcal{V} = \{\underline{v} \in H^1(\Omega)^d \quad \underline{v} = 0 \text{ on } \Gamma^D\} = \mathrm{H}^1_0(\Omega)$. after integrating equation over the whole domain Ω.

$$-\int_\Omega \underline{\nabla} \cdot \underline{\underline{\sigma}}_S(\underline{\mathbf{u}}) \cdot \underline{v}\, dx = \int_\Omega \underline{f} \cdot \underline{v}\, dx,$$

so that

$$\int_\Omega \underline{\underline{\sigma}}_S(\underline{\mathbf{u}}) \cdot \underline{\nabla}\, \underline{v}\, dx = \int_\Omega \underline{f} \cdot \underline{v}\, dx + \int_{\Gamma^N} \underline{g} \cdot \underline{v}\, d\Gamma,$$

and by symmetry of the Cauchy stress Tensor, we get

$$\int_\Omega \underline{\underline{\sigma}}_S(\underline{\mathbf{u}}) : \underline{\underline{\varepsilon}}(\underline{v})\, dx = \int_\Omega \underline{f} \cdot \underline{v}\, dx + \int_{\Gamma^N} \underline{g} \cdot \underline{v}\, d\Gamma.$$

Then

$$\int_\Omega \underline{\underline{\sigma}}_S(\underline{\mathbf{u}}) : \underline{\underline{\varepsilon}}(\underline{v})\, dx = \int_\Omega 2\mu\, \underline{\underline{\varepsilon}}(\underline{\mathbf{u}}) : \underline{\underline{\varepsilon}}(\underline{v}) dx + \int_\Omega \lambda\, \underline{\nabla}\cdot(\underline{\mathbf{u}})\, \underline{\nabla}\cdot(\underline{v}) dx = \int_\Omega \underline{f}\cdot\underline{v}\, dx + \int_{\Gamma^N} \underline{g}\cdot\underline{v}\, d\Gamma$$

Nearly Incompressible Linear Elasticity Formulation

In nearly incompressible material, ie the Poisson ratio ν approaches $1/2$ and λ/μ approaches infinity, in such a case the finite element discretization of the classical variational formulation will increasingly suffer from a locking phenomenon.

As in [BF91], the proposed remedy is based on introducing a new unknown $p = -\lambda \underline{\nabla} \cdot (\underline{\mathbf{u}})$ which is referred to as the pressure and living in a space $\mathcal{Q} \subset L^2(\Omega)$, with the later variable we get the following mixed formulation.

Find $(\underline{\mathbf{u}}, p) \in \mathcal{V} = \mathcal{M} \times \mathcal{Q}$ such that for all $(\underline{v}, q) \in \mathcal{V} = \mathcal{M} \times \mathcal{Q}$

$$\begin{cases} 2\displaystyle\int_\Omega \mu \underline{\underline{\varepsilon}}(\underline{\mathbf{u}}) : \underline{\underline{\varepsilon}}(\underline{v}) dx & - \displaystyle\int_\Omega p \underline{\nabla} \cdot (\underline{v}) dx = \int_\Omega \underline{f}\underline{v} dx + \int_{\Gamma^N} \underline{g} \cdot \underline{v}, \\ -\displaystyle\int_\Omega \underline{\nabla} \cdot (\underline{\mathbf{u}}) q dx & - \displaystyle\int_\Omega \frac{1}{\lambda} pq = 0. \end{cases} \tag{4.49}$$

FE discretization To discretize the problem (4.49), we have to do a good choice of the finite element spaces (for displacement and pressure) they have to satisfy ideally an inf-sup condition or at least to realize stability (see [BS08]).

Among these different spaces, the Taylor Hood mixed finite elements $\mathbb{P}_p - \mathbb{P}_{p-1}$ with $p > 0$ realize the inf-sup condition. Indeed, we will consider

$$\mathcal{M}_h = \mathbb{P}_p^d \text{ for the admissible space of displacement}$$

And

$$\mathcal{Q}_h = \mathbb{P}_{p-1} \text{ for the pressure}$$

So the discretized problem reads (in $\mathcal{M}_h = \mathcal{V}_h \times \mathcal{Q}_h$)

Find $(\underline{\mathrm{u}}_h, p_h) \in \mathcal{V}_h = \mathcal{M}_h \times \mathcal{Q}_h$ such that for all $(\underline{v}_h, q_h) \in \mathcal{V}_h$

$$\begin{cases} 2\displaystyle\int_\Omega \mu \underline{\underline{\varepsilon}}(\underline{\mathrm{u}}_h) : \underline{\underline{\varepsilon}}(\underline{v}_h) dx & - \displaystyle\int_\Omega p_h \underline{\nabla} \cdot (\underline{v}_h) dx = \int_\Omega \underline{f} \underline{v}_h dx + \int_{\Gamma^N} \underline{g}_h \cdot \underline{v}_h \\ - \displaystyle\int_\Omega \underline{\nabla} \cdot (\underline{\mathrm{u}}_h) q_h dx & - \displaystyle\int_\Omega \frac{1}{\lambda} p_h q_h = 0 \end{cases} \tag{4.50}$$

Then the derived linear system of the discrete formulation (4.50) reads

$$\mathbf{AU} = \begin{bmatrix} H & K^T \\ K & -C \end{bmatrix} \begin{bmatrix} \underline{\mathrm{u}} \\ p \end{bmatrix} = \begin{bmatrix} \mathrm{f} \\ 0 \end{bmatrix} = \mathrm{F}. \tag{4.51}$$

Given a basis $\phi_{j\,1 \leqslant j \leqslant n_1}$ for the admissible space $\mathcal{M}_h$ and $\psi_{j\,1 \leqslant n2}$ for $\mathcal{Q}_h$. The coefficients in the matrices are

$$H_{ij} = \int_\Omega 2\mu \underline{\underline{\varepsilon}}(\phi_j) : \underline{\underline{\varepsilon}}(\phi_i), \quad K_{ij} = -\int_\Omega \mathrm{div}(\phi_j)\psi_i, \quad C_{ij} = \int_\Omega \frac{1}{\lambda}\psi_j\psi_i.$$

Matrix $\widetilde{A}_i$ arises from the variational formulation (4.50) where the integration over domain Ω is replaced by the integration over subdomain Ω_i and finite element space $\mathcal{V}_h$ is restricted to subdomain Ω_i. Matrix B_i corresponds to a Robin problem and is the sum of matrix $\widetilde{A}_i$ and of the matrix of the following variational formulation restricted to the same finite element space:

$$\int_{\partial\Omega_i \setminus \partial\Omega} \frac{2\alpha\mu(2\mu+\lambda)}{\lambda+3\mu} \boldsymbol{u}_h \cdot \boldsymbol{v}_h \text{ with } \alpha = 10 \text{ in our test.} \tag{4.52}$$

In the next section, we explain the origin of the term (4.52).

4.6.2 Interface conditions

As it was studied in chapter 3, the term (4.52) is used so as to optimize the P.L Lions algorithm with the Robin conditions. We recall that to get it, we went through a Fourier study in a simple 2D situation.
In our case, let the global domain Ω be the whole plane $\mathbb{R}^2$ decomposed into two half planes $\Omega_1 := (-\infty, \delta) \times \mathbb{R}$ and $\Omega_2 := (0, \infty) \times \mathbb{R}$ where $\delta \geq 0$ is the width of the overlap, k denote the Fourier transform in the y direction, then the study yields to the following

$$\underline{\underline{\boldsymbol{\sigma}}} \cdot \underline{\boldsymbol{n}} + \mathcal{F}^{-1}\left(\begin{bmatrix} \frac{2|k_0|\mu(2\mu+\lambda)}{\lambda+3\mu} & 0 \\ 0 & \frac{2|k_0|\mu(2\mu+\lambda)}{\lambda+3\mu} \end{bmatrix} \begin{bmatrix} \mathcal{F}(\mathrm{u}_x) \\ \mathcal{F}(\mathrm{u}_y) \end{bmatrix}\right)$$

which simplifies in:

$$\underline{\underline{\boldsymbol{\sigma}}} \cdot \underline{\boldsymbol{n}} + |k_0| \frac{2\mu(2\mu+\lambda)}{\lambda+3\mu} \boldsymbol{u}\,, \tag{4.53}$$

where $\underline{\underline{\boldsymbol{\sigma}}}\,\underline{\boldsymbol{n}}$ is the normal stress vector, the velocity is decomposed into its normal u_x and tangential component u_y $\boldsymbol{u} = [\mathrm{u}_x,\, \mathrm{u}_y]^T$ and $\mathcal{F}$ denotes the Fourier transform in the y direction.

The interface condition (4.53) can be used for arbitrary domain decompositions since its variational formulation is the one of a stress free BVP to which we add the variational formulation of (4.52) where $\alpha := |k_0|$ for some chosen Fourier number k_0. Thus although the Fourier analysis has a limited domain of validity, the interface condition (4.53) can be used for arbitrary domain decompositions.

4.6.3 Eigenmodes incorporated in the Coarse Space

Elasticity In figures 4.2 and 4.1, we plot the eigenvectors of the generalized eigenvalue problems (4.32) and (4.30) for the linear elasticity case on the computational domain Ω see figure 4.5 . The domain decomposition is such that all subdomains contain the 8 alternating layers of steel and rubber. The GenEO coarse space for lower bound (Fig. 4.2) will consist of the first 12 modes. The first three are known as the rigid body modes. The other nine eigenmodes display very different behaviors for the steel and the rubber. The the 13th eigenvalue and the next ones are larger than 0.25 and are not incorporated into the coarse space. Interestingly enough, steel and rubber have the same deformations in these modes.

Stokes The same visualization of eigen modes was done in the case of Stokes problem with an homogeneous fluid. In figure 4.4 we can observe for the lower bound the existence of gap after the 2nd modes, $\lambda^2 = 10^{-13}$ and $\lambda^3 = 0.77$ which leads us to take these vectors preceding the gap in the coarse space.

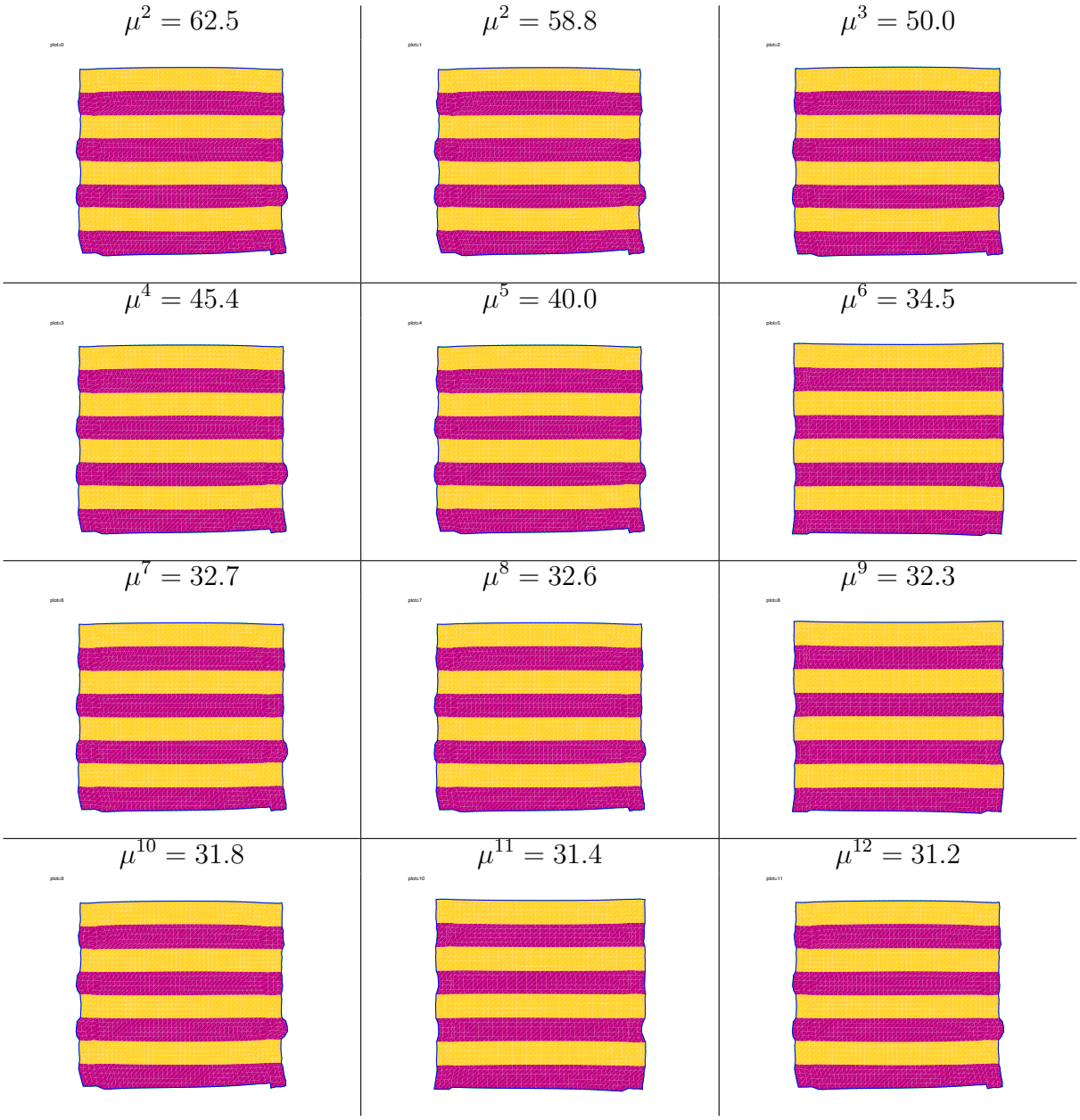


Fig 4.1 – Largest eigenvalues and corresponding eigenmodes of the GenEO II generalized eigenproblem for the upper bound–Elasticity equation on nearly-incompressible heterogeneous material (rubber and steel)

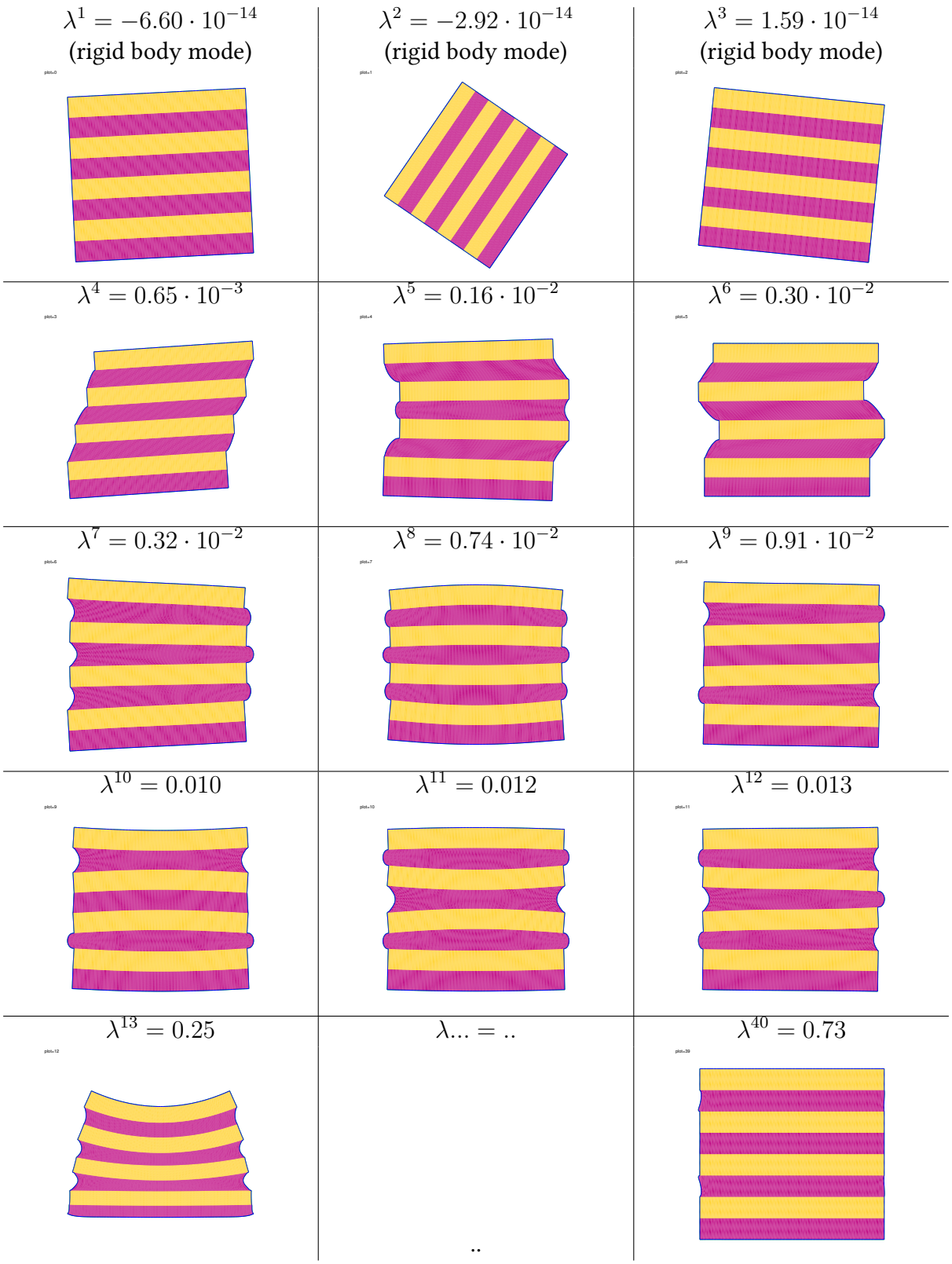


Fig 4.2 – Eigenvalues and eigenmodes of the GenEO II generalized eigenproblem for the lower eigenvalue (lower bound)–Elasticity equation on nearly-incompressible heterogeneous material (rubber and steel)

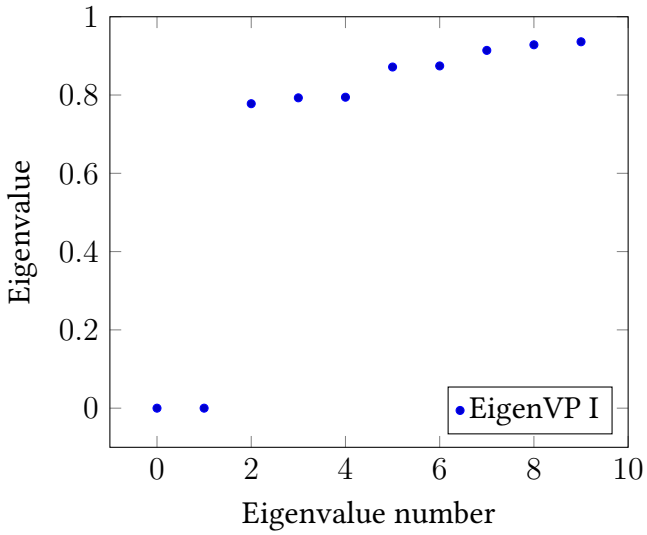


Fig 4.3 – Eigenvalues of the GenEO-2 eigenproblems (floating subdomain) Stokes problem.

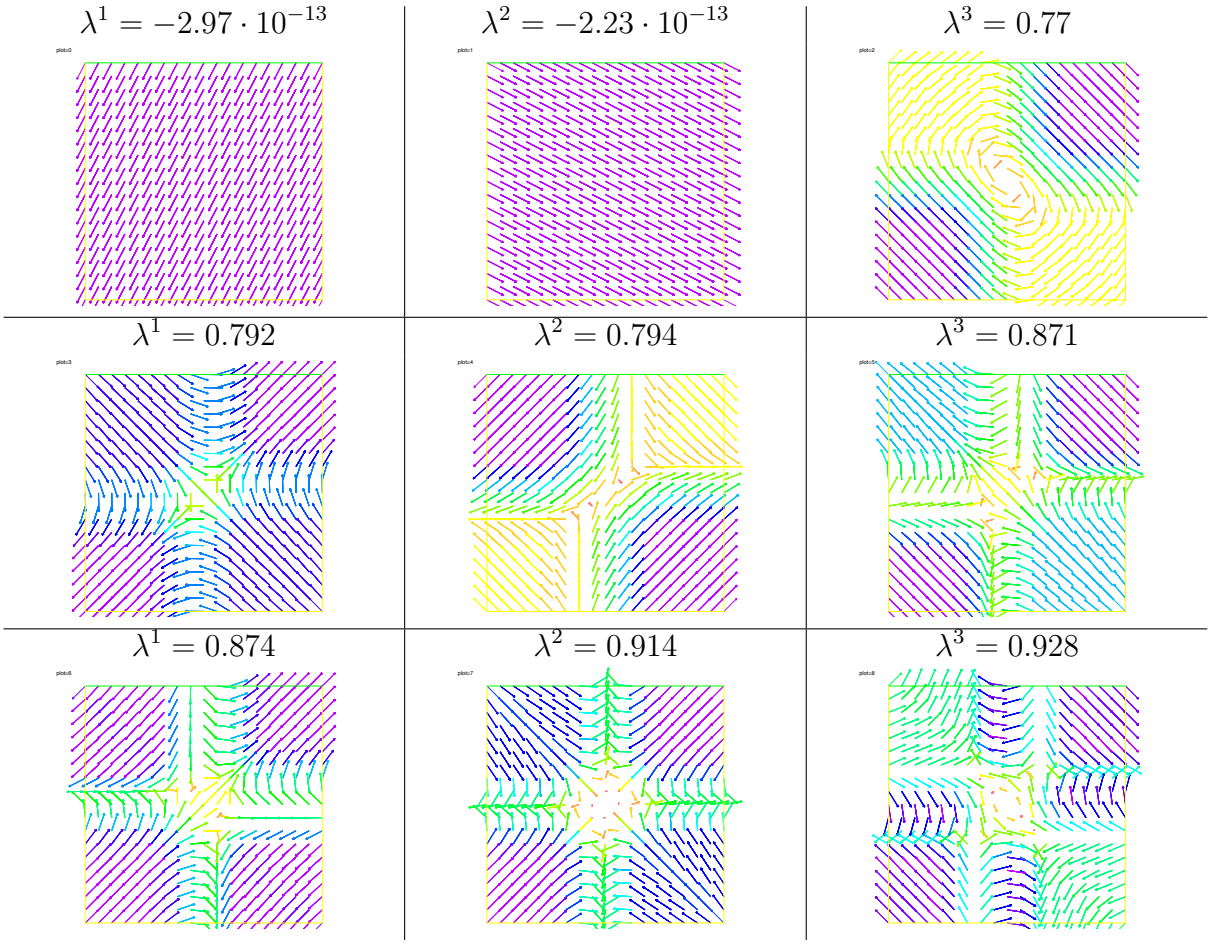


Fig 4.4 – Eigenvalues and eigenmodes of the GenEO II generalized eigenproblem for the lower bound –Stokes equation

Fig 4.5 – 2D Elasticity: coefficient distribution of steel and rubber.

		AS	SORAS	AS+ZEM		SORAS +ZEM		AS-GenEO		SORAS GenEO2	
d.o.f.	N	iter	iter	iter	*dim*	iter	*dim*	iter	*dim*	iter	*dim*
35841	8	150	184	117	*24*	79	*24*	110	*184*	13	*145*
70590	16	276	337	170	*48*	144	*48*	153	*400*	17	*303*
141375	32	497	>1000	261	*96*	200	*96*	171	*800*	22	*561*
279561	64	>1000	>1000	333	*192*	335	*192*	496	*1600*	24	*855*
561531	128	>1000	>1000	329	*384*	400	*384*	>1000	*2304*	29	*1220*
1077141	256	>1000	>1000	369	*768*	>1000	*768*	>1000	*3840*	36	*1971*

Table 4.1 – 2D Elasticity. GMRES iteration counts for a solid made of steel and rubber.

4.7 Numerical results

The new coarse space was tested quite successfully on nearly incompressible elasticity and Stokes problems with a discretization based on saddle point formulations in order to avoid locking phenomena.

Tests against other algorithms

We first report 2D results for a heterogeneous beam of eight layers of steel $(E_1, \nu_1) = (210 \cdot 10^9, 0.3)$ and rubber $(E_2, \nu_2) = (0.1 \cdot 10^9, 0.4999)$, see Figure 4.5. The beam is clamped on its left and right sides. Simulations were made with FreeFem++ [Hec12]. Iteration counts for various domain decomposition methods for a relative tolerance of 10^{-6} are given in Table 4.1. We compare the one level Additive Schwarz (AS) and SORAS methods, the two level AS and SORAS methods with a coarse space consisting of rigid body motions which are zero energy modes (ZEM) and finally AS with a GenEO coarse space as defined in [SDHNPS14] and SORAS with the GenEO-2 coarse space defined in Definition 22 with $\tau = 0.4$ and $\gamma = 10^3$. Columns *dim* refer to the total size of the coarse space of a two-level method. Eigenvalue problem (4.30) accounts for roughly 90% of the GenEO-2 coarse space size. We see that only the last method scales well with respect to the number of subdomains denoted by N.

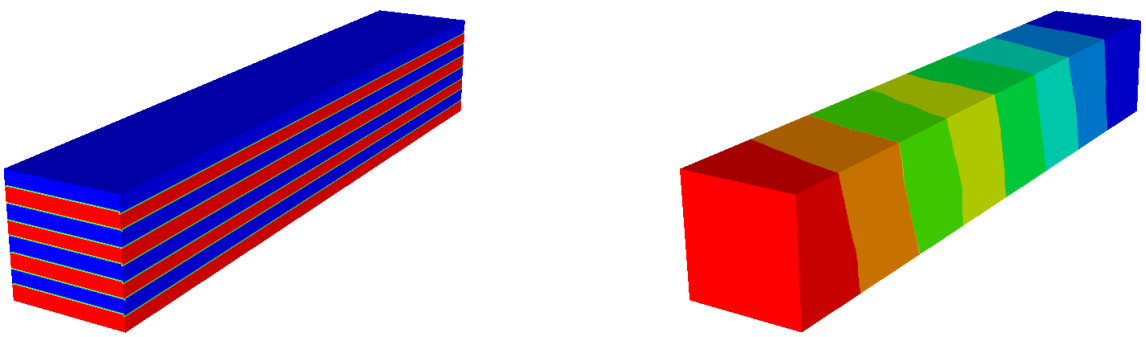


Fig 4.6 – Material coefficient, alternating layers of steel and rubber (left) and domain decomposition into 8 subdomains with a graph partitioner (right)

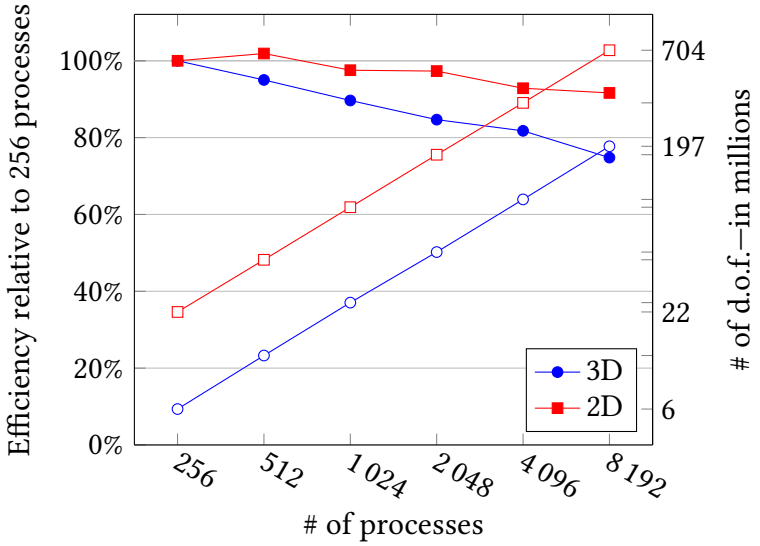


Fig 4.7 – Weak scaling experiments.

3D and 2D highly heterogeneous linear elasticity equations

Throughout this section we look at a linear elasticity problem with highly heterogeneous Lamé coefficients corresponding to steel and rubber materials. In the case of rubber which is nearly incompressible material the Poisson ratio ν approaches $1/2$ and $\lambda/\mu = 2\nu/(1-2\nu)$ approaches infinity. In order to avoid the resulting locking phenomena with finite element discretization, the pure displacement problem is replaced by a mixed formulation as proposed in [BS08]. We performed a large 2D and 3D simulations, on an heterogeneous beam, where the Lamé (E, ν) vary discontinuously over the domain in eight alternating layers of steel material with $(E_1, \nu_1) = (210 \times 10^9, 0.3)$ and rubber material with $(E_2, \nu_2) = (0.1 \times 10^9, 0.4999)$ submitted to an external forces, see Figure 4.6. The system is discretized using a Taylor-Hood mixed finite element discretization which are inf-sup stable. P_3/P_2 for the 2D case and P_2/P_1 for the 3D case.

The problem is solved with a minimal geometric overlap of one mesh element and a preconditioned GMRES is used to solve the resulting linear system where the stopping criteria for the relative residual norm is fixed to 10^{-6}. All the test cases were performed inside FreeFem++ code interfaced with the domain decomposition library HPDDM [JHNP13; JN14]. The factorizations are computed for each local problem and also for the global coarse problem using MUMPS [ADLK01]. Generalized eigenvalue problems to generate the GenEO space are solved using ARPACK [LSY98]. The coarse space is formed only with the generalized eigenvalue problem (4.30) since we noticed that the second one (4.32) has only a little effect on the convergence. All the results of this section were obtained using the super computer of IDRIS (institute of developpement of scientific computing resources) the machine is called Turing which is an IBM/Blue Gene/Q machine composed of 1024 compute nodes where each one is made of 16 cores PowerPC A2 clocked at 1.6 GHz.

These computations, see Figure 4.7, assess the weak scalability of the algorithm with respect to the problem size and the number of subdomains. All times are wall clock times. The domain is decomposed automatically into subdomains with a graph partitioner, ranging from 256 subdomains to 8192. and the problem size is increased by mesh refinement. In 3D the initial problem is about 6 millions d.o.f decomposed into 256 subdomains and solved in 145.2s and the final

	N	Factorization	Deflation	Solution	# of it.	Total	# of d.o.f.
3D	256	25.2 s	76.0 s	37.2 s	46	145.2 s	$6.1 \cdot 10^6$
	512	26.5 s	81.1 s	39.8 s	47	155.1 s	$12.4 \cdot 10^6$
	1 024	29.2 s	82.6 s	41.7 s	45	165.5 s	$25.0 \cdot 10^6$
	2 048	26.9 s	83.5 s	46.3 s	47	171.0 s	$48.8 \cdot 10^6$
	4 096	28.3 s	88.8 s	54.5 s	53	177.7 s	$97.9 \cdot 10^6$
	8 192	29.0 s	78.3 s	79.8 s	60	196.1 s	$197.6 \cdot 10^6$
2D	256	4.8 s	72.9 s	39.9 s	46	123.9 s	$22.1 \cdot 10^6$
	512	4.7 s	65.9 s	45.0 s	51	121.3 s	$44.0 \cdot 10^6$
	1 024	4.8 s	70.0 s	46.1 s	51	127.0 s	$88.3 \cdot 10^6$
	2 048	4.8 s	69.0 s	46.5 s	51	127.4 s	$176.8 \cdot 10^6$
	4 096	4.8 s	65.8 s	52.8 s	56	132.6 s	$351.0 \cdot 10^6$
	8 192	4.8 s	65.4 s	53.0 s	54	134.8 s	$704.1 \cdot 10^6$

Fig 4.8 – Weak scaling experiments elasticity timings tab .

problem is about 197 millions of d.o.f decomposed into 8192 subdomains and solved in 196s which gives an efficiency near to 75%. For the 2D case, the initial problem is approximatively of size 22 millions unknowns (d.o.f) decomposed into 256 subdomains and solved in 123.9s and we end up with a bigger problem about 704 millions unknowns (d.o.f) decomposed into 8192 subdomains and solved in 134s. The efficiency is close to 90%. In figure table 4.8, we report the number of GMRES iterations. Thanks to the robustness of the two-level domain decomposition preconditioner we can observe that they are quite good stable. We report in the same table all the timings concerning the algorithm, column "Factorization" concerns the local subdomains, the assembling and the factorization of the coarse operator are in column "Deflation" and in column "Solution" we display the time spent by GMRES.

4.7.1 3D and 2D incompressible Stokes system

Using the same libraries, we also performed a strong scaling test for an incompressible Stokes system of equations for a driven cavity problem:

Find $(\underline{\mathrm{u}}, p) \in H(\Omega)^{d=2,3} \times L^2_0(\Omega)$ such that

$$-\underline{\nabla} \cdot \underline{\underline{\sigma}}_F(\underline{\mathrm{u}}, p) = 0, \quad \text{and } \underline{\nabla} \cdot (\underline{\mathrm{u}}) = 0 \quad \text{in } \Omega, \tag{4.54}$$

with

$$\begin{cases} \underline{\underline{\sigma}}_F(\underline{\mathrm{u}}, p) = -pI + 2\mu\underline{\underline{\varepsilon}}(\underline{\mathrm{u}}), \\ \underline{\underline{\varepsilon}}(\underline{\mathrm{u}}) = \dfrac{1}{2}(\underline{\underline{\nabla}}\,\underline{\mathrm{u}} + (\underline{\underline{\nabla}}\,\underline{\mathrm{u}})^T) \text{ and } \varepsilon_{i,j} = \dfrac{1}{2}\left(\dfrac{\partial \mathrm{u}_i}{\partial x_j} + \dfrac{\partial \mathrm{u}_j}{\partial x_i}\right). \end{cases} \tag{4.55}$$

As a boundary conditions, we consider a continuous velocity on the upper face and zero Dirichlet otherwise. The computations are done in both two and three dimensions on a domain $\Omega = [0,1]^2$ and $\Omega = [0,1]^3$, respectively. Once more the problems are discretized via Taylor-Hood finite element P_2/P_1 with a continuous pressure.

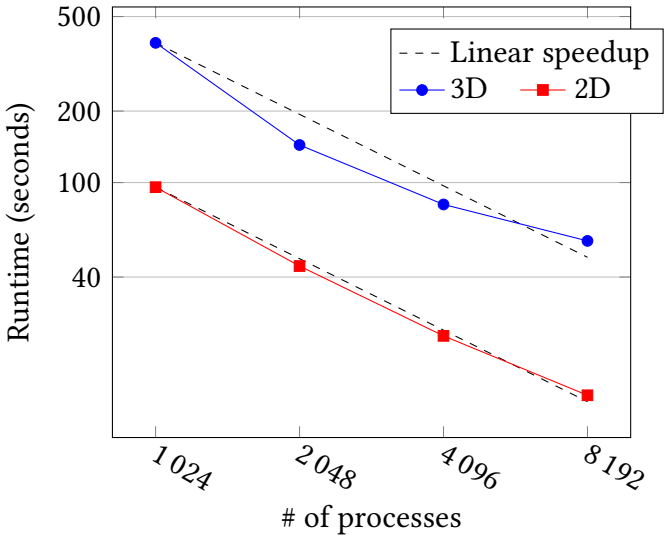


Fig 4.9 – Timings of various simulations Stokes.

	N	Factorization	Deflation	Solution	# of it.	Total	# of d.o.f.
3D	1 024	79.2 s	229.0 s	76.3 s	45	387.5 s	$50.63 \cdot 10^6$
	2 048	29.5 s	76.5 s	34.8 s	42	143.9 s	
	4 096	11.1 s	45.8 s	19.8 s	42	80.9 s	
	8 192	4.7 s	26.1 s	14.9 s	41	56.8 s	
2D	1 024	5.2 s	37.9 s	51.5 s	51	95.6 s	$100.13 \cdot 10^6$
	2 048	2.4 s	19.3 s	22.1 s	42	44.5 s	
	4 096	1.1 s	10.4 s	10.2 s	35	22.6 s	
	8 192	0.5 s	4.6 s	6.9 s	38	12.7 s	

(a) Breakdown of the timings used for the figure on top

Fig 4.10 – Strong scaling experiments.

We assess here the strong scalability of the algorithm. For this, we make the number of subdomains vary for a fixed global system size. In our test case the system size is fixed to 50 millions unknowns (d.o.f) in 3D and to 100 millions unknowns (d.o.f) in 2D, as we can show in figure 4.9, from 1024 subdomains to 8192 subdomains we get a quite good speed up. In the three dimensional case, we pass from 387.5s using 1024 subdomains to 56.8s when using 8192 subdomains. In figure table 4.10 we display all timings relative to this test, column "Factorization" gives the time spent in the factorization of the local submatrices, column "Deflation" corresponds to local eigenvalue solvers and the coarse space correction construction, column "Solution" is the time taken by the GMRES solve of the global linear system by the domain decomposition algorithm.

4.8 Conclusion

We developed a theory for the overlapping P.L. Lions' algorithm similar to the existing one for the Schwarz algorithm in that we show how to build adaptively a coarse space so that the two-level preconditioner achieves a targeted condition number. The theory is based on the introduction of the SORAS (4.14) algorithm which is a new symmetric variant of the ORAS

preconditioner. The two-level method is implemented in the HPDDM library that is interfaced with finite element solvers such as FreeFem++ and Feel++.

Note that for a given targeted condition number, the size of the coarse space depends on the interface condition. A small coarse space is important in order to achieve good scalability results. Thus, it might be interesting to optimize this condition with respect to the coarse space size.

Chapter 5

Application to Navier-Stokes equations

Contents

5.1 Introduction

In this chapter we present the application of the previously (4.14) designed Optimized Schwarz domain decomposition preconditioner in chapter 4 to the unsteady Navier-Stokes equations. Domain decomposition methods follow a paradigm which yields a natural parallelization, this is why we will rely on DDM strategy in order to solve Navier-Stokes equation in a parallel framework and validate a large scale simulation of Navier-Stokes equations on a known benchmark [STDKR96].

5.2 Navier-Stokes equations

Let Ω be a bounded domain of $\mathbb{R}^3$, $\Gamma = \partial\Omega$ its boundary, we denote Γ_D the subset of Γ where the essential (Dirichlet) boundary condition is applied, the Neumann boundary condition can be considered on $\Gamma_N = \Gamma \backslash \Gamma_D$, we denote by $[0, T]$ the time interval. The strong form of the incompressible Navier-Stokes equations read:

Find the velocity $\underline{\mathrm{u}}$ and the pressure p such that

$$\begin{cases} \rho\dfrac{\partial \underline{\mathrm{u}}}{\partial t} + \rho\,\underline{\mathrm{u}}\cdot\nabla\underline{\mathrm{u}} - \underline{\nabla}\cdot\underline{\underline{\sigma}}_F(\underline{\mathrm{u}},p) = \underline{\mathrm{f}} \text{ in } \Omega\times(0,T) \\ \underline{\nabla}\cdot\underline{\mathrm{u}} = 0 \text{ in } \Omega\times(0,T) \\ \underline{\mathrm{u}} = \underline{\mathrm{g}} \text{ on } \Gamma_D\times(0,T) \\ \underline{\underline{\sigma}}_F(\underline{\mathrm{u}},p)\underline{n} = \mathrm{h} \text{ on } \Gamma_N\times(0,T) \\ \underline{\mathrm{u}}(0) = \mathrm{u}_0 \text{ in } \Omega\times 0 \end{cases} \tag{5.1}$$

with $\underline{\mathrm{f}}$ a given force vector, μ the kinematic viscosity, ρ is the density and $\underline{n}$ is the outward normal direct unit normal vector to Γ_N. With a chosen characteristic velocity U_0, we define $Re = U_0/\mu$, the Reynolds number a dimensionless coefficient that characterizes to nature of de flow described by Navier-Stokes equations.

The stress tensor $\underline{\underline{\sigma}}_F$ for a Newtonian fluid is defined by

$$\underline{\underline{\sigma}}_F = -pI + 2\mu\underline{\underline{\varepsilon}}(\underline{\mathrm{u}}) \quad \text{and} \quad \underline{\underline{\varepsilon}}(\underline{\mathrm{u}}) = \frac{1}{2}\left(\underline{\underline{\nabla}}\,\underline{\mathrm{u}} + (\underline{\underline{\nabla}}\,\underline{\mathrm{u}})^T\right).$$

We recall that the equations represent the balance of momentum, the mass conservation, the essential and natural boundary conditions and the initial condition.

In order to do a spatial approximation of the Navier-Stokes equations (5.1), we consider the two following Hilbert spaces $\mathcal{V} = \{\underline{v} \in H^1(\Omega)^d \quad \underline{v} = g \text{ on } \Gamma^D\}$ and $\mathcal{Q} = L^2(\Omega)$,
Then after integrating over the whole domain Ω, the variational formulation of Navier-Stokes equation reads for all $t \in [0,T]$(see [QV08])

Find $U(t) = (\underline{\mathrm{u}}, p) \in \mathcal{V}\times\mathcal{Q}$ such that for all $V(t) = (\underline{v}, q) \in \mathcal{V}\times\mathcal{Q}$

$$\begin{cases} \displaystyle\int_\Omega \rho\,\frac{\partial \underline{\mathrm{u}}}{\partial t}\,\underline{v}dx + \int_\Omega \rho\,\underline{\mathrm{u}}\cdot\nabla\underline{\mathrm{u}}\,\underline{v}dx + \int_\Omega 2\mu\underline{\underline{\varepsilon}}(\underline{\mathrm{u}}) : \underline{\underline{\varepsilon}}(\underline{v})dx \\ \displaystyle -\int_\Omega p\,\underline{\nabla}\cdot(\underline{v})dx = \int_\Omega \underline{f}\,\underline{v}\,dx + \int_{\Gamma^N}\underline{g}\cdot\underline{v}\,d\sigma, \\ \displaystyle\int_\Omega \underline{\nabla}\cdot(\underline{\mathrm{u}})\,q\,dx = 0. \end{cases}$$

Let us introduce at this point a suitable finite element space. We denote by $\mathcal{V}_h \in \mathcal{V}$ the chosen conforming space of finite element functions for velocity , $\mathcal{V}_h$ a set of vector functions and $\mathcal{Q}_h \in \mathcal{Q}$ the chosen conforming space of finite element functions for pressure.

$$\mathcal{V}_h = \{\underline{v}_h \in C^0(\overline{\Omega})^{2,3} \quad | \quad \underline{v}_{h_{|T}} \in \mathcal{P}^k, \forall T \in \mathcal{T}_h\},$$
$$\mathcal{Q}_h = \{\underline{v}_h \in C^0(\overline{\Omega}) \quad | \quad \underline{v}_{h_{|T}} \in \mathcal{P}^{k-1}, \forall T \in \mathcal{T}_h\},$$

where $\mathcal{P}^k$ is the space of polynomials of degrees less than or equal to k, and $\mathcal{T}_h$ is the considered triangulation of the domain Ω. as we are dealing with a mixing formulation on velocity and pressure, the spatial discretization have to satisfy an inf-sup condition, this why Taylor-Hood finite element space $\mathcal{P}^k - \mathcal{P}^{k-1}$ (e.g. $\mathcal{P}^2 - \mathcal{P}^1$) will be our choice here.

Time discretization We choose to do the time discretization of the Navier-Stokes equations (5.1) by means of Backward Differentiation Formula (BDF) [Sag06] which are a family of multistep methods. These high order schemes are often used in computational fluid dynamics, see for example [FD15; GSV06].

First, we subdivide the time-interval $[0, T]$ into N_t subdivision of equal size $\Delta t = \dfrac{T}{N_t}$, where $t_n = n\Delta t$ for $n = 0, ..., N_t$ is the time instance

$$\frac{\partial \underline{\mathbf{u}}_h}{\partial t} = \frac{\gamma \underline{\mathbf{u}}_h^{n+1} - \underline{\mathbf{u}}_h^{n,BDF\gamma}}{\Delta t}, \tag{5.2}$$

where the BDF scheme of orders $\gamma = 1, 2, 3$ reads

$$\underline{\mathbf{u}}_h^{n,BDF\gamma} = \begin{cases} \underline{\mathbf{u}}_h^n & \text{if } n \geqslant 0, \quad \text{for BDF1} \\ 2\underline{\mathbf{u}}_h^n - \frac{1}{2}\underline{\mathbf{u}}_h^{n-1} & \text{if } n \geqslant 0, \quad \text{for BDF2} \\ 3\underline{\mathbf{u}}_h^n - \frac{3}{2}\underline{\mathbf{u}}_h^{n-1} + \frac{1}{3}\underline{\mathbf{u}}_h^{n-2} & \text{if } n \geqslant 0, \quad \text{for BDF3} \end{cases}$$

and

$$\gamma = \begin{cases} 1 & \text{for BDF1} \\ \frac{3}{2} & \text{for BDF2} \\ \frac{11}{6} & \text{for BDF3.} \end{cases}$$

Then the fully discrete Navier-Stokes problem (5.1) reads:
Find $U_h(t_{n+1}) = (\underline{\mathbf{u}}_h^{n+1}, p_h^{n+1}) \in \mathcal{V}_h \times \mathcal{Q}_h$ such that for all $V_h(t) = (\underline{v}_h, q_h) \in \mathcal{V}_h \times \mathcal{Q}_h$

$$\begin{cases} \displaystyle\int_\Omega \rho \frac{\gamma \underline{\mathbf{u}}_h^{n+1}}{\Delta t}, \underline{v} dx + \int_\Omega \rho \boxed{\underline{\mathbf{u}}_\mathbf{h}^{\mathbf{n+1}} \cdot \nabla \underline{\mathbf{u}}_\mathbf{h}^{\mathbf{n+1}}} \underline{v}_h dx + 2 \int_\Omega \mu \underline{\underline{\varepsilon}}(\underline{\mathbf{u}}_h^{n+1}) : \underline{\underline{\varepsilon}}(\underline{v}) dx \\ \displaystyle - \int_\Omega p_h^{n+1} \underline{\nabla} \cdot (\underline{v}_h) dx = \int_\Omega \underline{f}\, \underline{v}\, dx + \int_{\Gamma^N} \underline{g} \cdot \underline{v}_h\, d\sigma + \int_\Omega \frac{1}{\Delta t} \underline{\mathbf{u}}_h^{n,BDF\gamma}, \\ \\ \displaystyle \int_\Omega \underline{\nabla} \cdot (\underline{\mathbf{u}}_h^{n+1})\, q_h\, dx = 0. \end{cases}$$

At each time-step the BDF scheme yields a non linear problem du to the term $\underline{\mathbf{u}}_h^{n+1} \cdot \nabla \underline{\mathbf{u}}_h^{n+1}$ to be solved, we can proceed to the approximation of this non-linearity by means of the Newton method as in [GR12; ESW14], at each time step it requires the assembly of the Jacobian matrix and a solution of several linear systems. This is called a fully implicit discretization approach, which has been widely used in the study of incompressible fluid flows both in laminar and turbulent regimes as in [KF04; UPDP11].

However, such a fully implicit approach requires a significant computational cost du to the multiple assembling of the right-hand side at each time step and the several linear system to solve in order to let Newton method converge for each iterate.

In order to reduce the cost of computation, one opts for a semi-implicit approach. To do this, we consider the following Newton-Gregory backward polynomials extrapolation of orders $\gamma = 1, 2, 3$ for the the velocity at the discrete time t_{n+1} see [CK06] for derivation detail.

$$\underline{\mathrm{u}}_h^{n+1}\cdot\nabla\underline{\mathrm{u}}_h^{n+1} = \begin{cases} \underline{\mathrm{u}}_h^{n}\cdot\nabla\underline{\mathrm{u}}_h^{n+1} & \text{if } n\geqslant 0, \quad \text{for BDF1} \\ \left(2\underline{\mathrm{u}}_h^{n}-\underline{\mathrm{u}}_h^{n-1}\right)\cdot\nabla\underline{\mathrm{u}}_h^{n+1} & \text{if } n\geqslant 0, \quad \text{for BDF2} \\ \left(3\underline{\mathrm{u}}_h^{n}-3\underline{\mathrm{u}}_h^{n-1}+\underline{\mathrm{u}}_h^{n-2}\right)\cdot\nabla\underline{\mathrm{u}}_h^{n+1} & \text{if } n\geqslant 0, \quad \text{for BDF3} \end{cases}$$

Then the fully discrete semi-implicit formulation of Navier-Sokes problem (5.1) reads for BDF with order 2

Find $U_h(t_{n+1}) = (\underline{\mathrm{u}}_h^{n+1}, p_h^{n+1}) \in \mathcal{V}_h \times \mathcal{Q}_h$ such that for all $V_h(t) = (\underline{v}_h, q_h) \in \mathcal{V}_h \times \mathcal{Q}_h$

$$\begin{cases} \displaystyle\int_\Omega \rho\tfrac{3}{2}\frac{\underline{\mathrm{u}}_h^{n+1}}{\Delta t}\,\underline{v}\,dx + \int_\Omega \rho\left(2\underline{\mathrm{u}}_h^{n}-\underline{\mathrm{u}}_h^{n-1}\right)\cdot\nabla\underline{\mathrm{u}}_h^{n+1}\,\underline{v}_h\,dx + 2\int_\Omega \mu\underline{\underline{\varepsilon}}(\underline{\mathrm{u}}_h^{n+1}) : \underline{\underline{\varepsilon}}(\underline{v})\,dx \\ \displaystyle -\int_\Omega p_h^{n+1}\,\underline{\nabla}\cdot(\underline{v}_h)\,dx = \int_\Omega \underline{f}\,\underline{v}\,dx + \int_{\Gamma^N}\underline{g}\cdot\underline{v}_h\,d\sigma - \int_\Omega \frac{\rho}{\Delta t}\left(2\underline{\mathrm{u}}_h^{n}-\tfrac{1}{2}\underline{\mathrm{u}}_h^{n-1}\,dx\right), \\ \displaystyle\int_\Omega \underline{\nabla}\cdot(\underline{\mathrm{u}}_h^{n+1})\,q_h\,dx = 0. \end{cases}$$

Thanks to the semi-implicit approach in the time discretization, the above scheme yields to one linear system in velocity $\underline{\mathrm{u}}_h^{n+1}$ and pressure p_h^{n+1} to be solved at each time step t_{n+1}.
The linear system to solve reads

$$\mathbf{A} = \begin{bmatrix} H & L^T \\ L & 0 \end{bmatrix}\begin{bmatrix} \boldsymbol{U^{n+1}} \\ \boldsymbol{P^{n+1}} \end{bmatrix} = \mathbf{F}, \tag{5.3}$$

with $H = \dfrac{1}{\Delta t}M + A + C(U^n)$, where M is the fluid mass matrix, A is the stiffness matrix, and $C(U^n)$ the (linearized) convection terms of the momentum equation, L and L^T are the discretized counterparts of divergence operator and gradient operator, $\boldsymbol{U^{n+1}}$ is the vector of velocity unknowns and $\boldsymbol{P^{n+1}}$ the one of pressure unknowns at time t_n. Note that due to the convective term, matrix H and thus $\mathbf{A}$ are not symmetric. Our parallel solver will be a one level Schwarz domain decomposition method.

5.3 Linear solver: Domain Decomposition Method

In this section, we recall the core domain decomposition method that will be used in order to perform simulations with the Navier-Stokes equations (5.1).

From the finite element mesh $\mathcal{T}_h$, we decompose the domain Ω into N overlapping subdomains $\Omega_i, \quad i = 1, ...N$, so that all subdomains are a union of cells of the mesh $\mathcal{T}_h$. This decomposition induces a natural decomposition of the set of indices $\mathcal{N}$ into N subsets of indices $(\mathcal{N}_i)_{1\leq i\leq N}$:

$$\mathcal{N}_i := \{k\in\mathcal{N} \mid meas(\text{supp}\,\phi_k \cap \Omega_i) > 0\}\,,\ 1\leq i\leq N. \tag{5.4}$$

For all $1 \leq i \leq N$, let R_i be the restriction matrix from $\mathbb{R}^{\#\mathcal{N}}$ to the subset $\mathbb{R}^{\#\mathcal{N}_i}$ and D_i be a diagonal matrix of size $\#\mathcal{N}_i \times \#\mathcal{N}_i$, which forms a partition of unity at the algebraic level,

$I_d = \sum_{i=1}^{N} R_i^T D_i R_i$, where $I_d \in \mathbb{R}^{\#\mathcal{N}\times\#\mathcal{N}}$ is the identity matrix.
For all subdomains $1 \leq i \leq N$, let B_i be a matrix of size $\#\mathcal{N}_i \times \#\mathcal{N}_i$, which comes from the local discretization of the boundary value local problems using optimized Robin transmission conditions, In the case of Navier-Stokes equations (5.1) the used optmized Robin interface condition reads

$$\underline{\underline{\sigma}}_F \underline{n} + \frac{2\alpha(\mu/Re)(2\mu/Re+\lambda)}{\lambda+3\mu/Re}\boldsymbol{u} = 0 \quad \text{on } \partial\Omega_i \setminus \partial\Omega, \tag{5.5}$$

with $\alpha = 10$ and $\lambda = 10^8$ in our test. Actually λ is chosen very large since we have an incompressible flow. The Robin parameter is thus very close to $2\alpha\,\mu/Re$. We recall the SORAS and ORAS preconditioners defined in chapter 4

$$M_{SORAS,1}^{-1} := \sum_{i=1}^{N} R_i^T D_i B_i^{-1} D_i R_i\,, \tag{5.6}$$

and the ORAS precosnditioner in his non symmetric version

$$M_{ORAS,1}^{-1} := \sum_{i=1}^{N} R_i^T D_i B_i^{-1} R_i\,. \tag{5.7}$$

Now we equip both SORAS preconditioner (5.6) and ORAS preconditioner (5.7) with recycling technic. The next section will be dedicated to this technics.

Remark 10
Our numerical experiments show us that we are already quite stable with one-level preconditioner and deflation technic when dealing with time dependent equation. This is why, there is no need to use two-level preconditioners.

5.4 Recycling Krylov subspaces

Performing one simulation in an interval of time $[0, T]$ requires the solves of a linear system at each time step t_n:

$$\mathbf{A}^n\mathbf{U}^n = \mathbf{F}^n \quad n = 1, 2, \ldots$$

We note that the multiple linear systems are slowly changing from one time step to an other. This leads us to use some recycling technics. As we are using domain decomposition method like a preconditioner for Krylov solver, it allows us to use and take advantage from the important existing works in the area of recycling Krylov subspaces.

Many works have been done on strategies of recycling of Krylov subspace during a long sequence of slowly changing linear systems (e.g. $A^i\mathrm{x}^i = b^i$, $i = 1, 2\ldots$), see [SO10; ARSO14; SYEG00]. This allows to reduce significantly the cost of solving the subsequent system. Among the existing works, we made the choice to use the GCRO-DR algorithm which was proposed by Parks et al. in [PSMJM06b] with the SORAS (5.6) and ORAS (5.7) DDM preconditiioners (5.6).

5.4.1 Generalized Conjugate Residual method with inner Orthogonalization and Deflated Restarting (GCRO-DR)

GCRO-DR algorithm is based on GMRES-DR [Mor02] and GCROT [DS99] in which we can find technics in retaining cleverly a selected subspace. In this section we present some details on the GCRO-DR algorithm, see Algorithm 4 for the detailed description.
In order to keep this part as self-contained as possible, we give a brief description of the GCRO-DR algorithm used in the library HPDDM [JHNP13; JN14]. The implementation detail is given in the paper of Jolivet et al. [JT16].

First, we introduce the following notations

- n is the size of all linear systems.
- m the maximum dimension of Krylov subspaces.
- k is the dimension of recycled Krylov subspace.
- V_{m+1} is an Arnoldi basis of $m+1$ vector of size n.
- $\underline{H}_m$ is the Hessenberg matrix.

$$H = H_m + QR^{-H} \begin{bmatrix} 0_{(m-1)\times(m-1)} & 0_{(m-1)\times 1} \\ 0_{1\times(m-1)}, & h_{m+1,m}^H h_{m+1,m} \end{bmatrix} \tag{5.8}$$

H is the Hessenberg matrix.
The generalised eigenvalue problem defined in line 32 -see Algorithm 4 is defined with the following matrices

$$\begin{aligned} T =& G_m^H G_m \\ W =& G_m^H \begin{bmatrix} G_k^H U_k & 0_{k\times(m-k)} \\ V_{m-k+1}^H U_k & I_{(m-k+1)\times(m-k)} \end{bmatrix}. \end{aligned} \tag{5.9}$$

where G_m is defined as

$$G_m = \begin{bmatrix} D_k & E_k \\ 0_{(m-k+1)\times k} & \overline{H}_{m-k} \end{bmatrix}.$$

5.5 Large-Scale simulation

Throughout this section, we conduct a large-scale experiement on the well-known three dimension benchmark problem of a vortex shedding induced by the fluid flow past a cylinder for a low Reynolds number. A detailed overview of benchmark computations of laminar flow around a cylinder is given in [STDKR96].

Flow past a cylinder In figure 5.1, we show the considered geometrical setting for the benchmark, where the computational Ω domain is the channel with a cylinder, The height of the channel is $H = 0.41m$ and the diameter of the cylinder is $D = 0.1m$.

We perform simulations for the 3D test case 3D-2Q in [STDKR96], where the right hand side

Algorithm 4 GCRO with deflated restarting (GCRO-DR)

$r_0 = b_i - Ax_0$
if U_k is defined (from solving a previous system) **then**
    $[Q, R] =$ distributed qr(AU_k)
    $C_k = Q$
    $U_k = U_k R^{-1}$
    $x_1 = x_0 + U_k C_k^H r_0$
    $r_1 = r_0 - C_k C_k^H r_0$
else
    $v_1 = r_0 / \|r_0\|_2$
    $c = \|r_0\|_2 e_1$
    perform m steps of GMRES, thus generating V_{m+1} and $[Q, R] =$ qr$(\overline{H}_m)$
    find Y_m such that $RY_m = Q^{-1} \begin{bmatrix} v_1 \\ 0_{(m-1)\times 1} \end{bmatrix}$
    $x_1 = x_0 + V_m Y_m$
    $r_1 = b - Ax_1$
    solve $Hz_\lambda = \theta_\lambda z_\lambda$ ▷ cf. eq. (5.8)
    store the k eigenvectors z_λ associated to the smallest eigenvalues in magnitude in P_k
    $[Q, R] =$ qr$(\overline{H}_m P_k)$
    $C_k = V_{m+1} Q$
    $U_k = V_m P_k R^{-1}$
end if
$j = 1$
while $(\|r_i\|_2 > \varepsilon)$ **do**
    $v_i = r_i / \|r_i\|_2$
    $c = \|r_i\|_2 e_i$
    $j \mathrel{+}= 1$
    perform $m - k$ steps of GMRES with the linear operator $(I - C_k C_k^H)A$, thus generating V_{m+1-k}, $[Q, R] =$ qr$(\overline{H}_{m-k})$, and $E_k = C_k A V_{m-k}$
    find Y_{m-k} such that $RY_{m-k} = Q^{-1} \begin{bmatrix} v_i \\ 0_{p\cdot(m-k-1)\times 1} \end{bmatrix}$
    $Y_k = C_k^H R_{j-1} - E_k Y_{m-k}$
    $x_j = x_{j-1} + U_k Y_k + V_{m-k} Y_{m-k}$
    $r_j = b_i - Ax_j$
    scale the columns of U_k so that they are of unit norm
    solve $Tz_\lambda = \theta_\lambda W z_\lambda$ ▷ cf. eq. (5.9)
    store the k eigenvectors z_λ associated to the smallest eigenvalues in magnitude in P_k
    $[Q, R] =$ qr$(\overline{H}_m P_k)$
    $C_k = \begin{bmatrix} C_k & V_{m-k+1} \end{bmatrix} Q$
    $U_k = \begin{bmatrix} U_k P_k & V_{m-k} P_k \end{bmatrix} R^{-1}$
end while

of the momentum equation vanishes i.e. $\mathrm{f} = 0$, the kinematic viscosity of the fluid is $\mu = 0.001m^2/s$ and its density is $\rho = 1kg/m^3$. The flow has the Reynolds number $Re = 100$.

As for the boundary conditions, at the surface on the top, bottom, lateral of the domain we set a null velocity $\underline{\mathrm{u}} = 0$. At the inflow of the domain Γ_{in} we prescribe the following dirichlet condition

$$U(0,x,y,z) = \begin{bmatrix} \frac{16}{H^4} U_m yz(H-y)(H-y) \\ 0 \\ 0 \end{bmatrix},$$

with $U_m = 2.25m/s$. At the outflow boundary of the domain Γ_{out} we consider the following natural boundary conditions

$$-p\overrightarrow{n} + 2\mu\left(\underline{\underline{\nabla}}\,\underline{\mathrm{u}} + (\underline{\underline{\nabla}}\,\underline{\mathrm{u}})^T\right)\cdot\overrightarrow{n} = 0. \quad \text{on } \Gamma_{out}$$

where $\overrightarrow{n}$ is the outward direct unit vector normal to Γ_{out}. The flow has the Reynolds number $Re = 100$.

In order to validate the Benchmark results with the results available in literature, we compute three important features for this flow, the drag, the lift coefficients at the cylinder and the Strouhal number as given in [STDKR96]

$$\begin{aligned} C_d(t) &:= \frac{2}{\rho D\overline{U}_{max}H}\oint_S \left(\rho\mu\frac{\partial\underline{\mathrm{u}}_t}{\partial n}n_y - pn_x\right)d\sigma \\ C_l(t) &:= -\frac{2}{\rho D\overline{U}_{max}H}\oint_S \left(\rho\mu\frac{\partial\underline{\mathrm{u}}_t}{\partial n}n_x - pn_y\right)d\sigma \end{aligned} \tag{5.10}$$

with the following notation: S is the surface of the cylinder, $\overrightarrow{n} = (n_x, n_y, nz)$ is the outward normal to S, $\underline{\mathrm{u}}_t$ the tangential velocity on S.

In Our simulation, the aerodynamic drag and lift coefficients have been computed by mean of weak residual form [JM01; CB16] which ensures higher accuracy than the direct use of equations (5.10).

$$\begin{aligned} C_d(t) &:= -\frac{2}{\rho D\overline{U}_{max}H}\left(\int_\Omega \rho\left(\frac{\partial\underline{\mathrm{u}}}{\partial t} + \underline{\mathrm{u}}\cdot\nabla\underline{\mathrm{u}}\right)\underline{v}_d + \mu\underline{\underline{\varepsilon}}(\underline{\mathrm{u}}) : \underline{\underline{\varepsilon}}(\underline{v}_d) - p\,\underline{\nabla}\cdot\underline{v}_d\,dx\right), \\ C_l(t) &:= -\frac{2}{\rho D\overline{U}_{max}H}\left(\int_\Omega \rho\left(\frac{\partial\underline{\mathrm{u}}}{\partial t} + \underline{\mathrm{u}}\cdot\nabla\underline{\mathrm{u}}\right)\underline{v}_l + \mu\underline{\underline{\varepsilon}}(\underline{\mathrm{u}}) : \underline{\underline{\varepsilon}}(\underline{v}_l) - p\,\underline{\nabla}\cdot\underline{v}_l\,dx\right), \end{aligned} \tag{5.11}$$

$\underline{v}_d$ and $\underline{v}_l$ are an appropriate defined dimensionless test function (see [JM01])

Numerical results We have implemented the semi-implicit scheme with BDF2 in time discretization using the open source finite element software FreeFem++ [Hec12] interfaced with the domain decomposition library HPDDM [JHNP13; JN14] both are under the LGPL license.

All computations are carried out using CURIE machine composed of 5040 nodes made of two

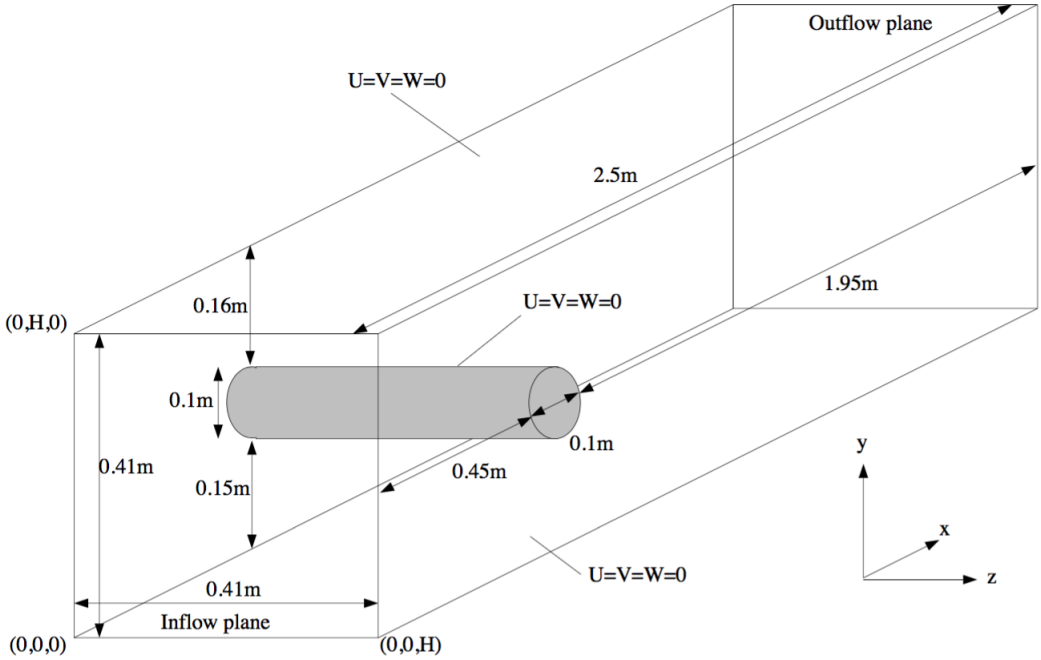


Fig 5.1 – Geometrical configuration for the flow around a cylinder with circular cross section.

eight-core Intel Sandy Bridge processors clocked at 2.7 Ghz with a peak performance of 1.7 PFLOP$/s$. The Math Kernel Library (MKL Intel) was used for the linear algebra operations for both dense and sparse computations. METIS [KK98] was used for graph partitioning. MUMPS [ADLK01] library was used for all local matrices factorization.
In each simulation we compute the Drag (C_d), lift(C_l) aerodynamics coefficients, and the Strouhal number, the result remains the same for all simulations where $C_d = 3.31$, $C_l = -0.01$ which is in agreement with the Turek benchmark results in [STDKR96].

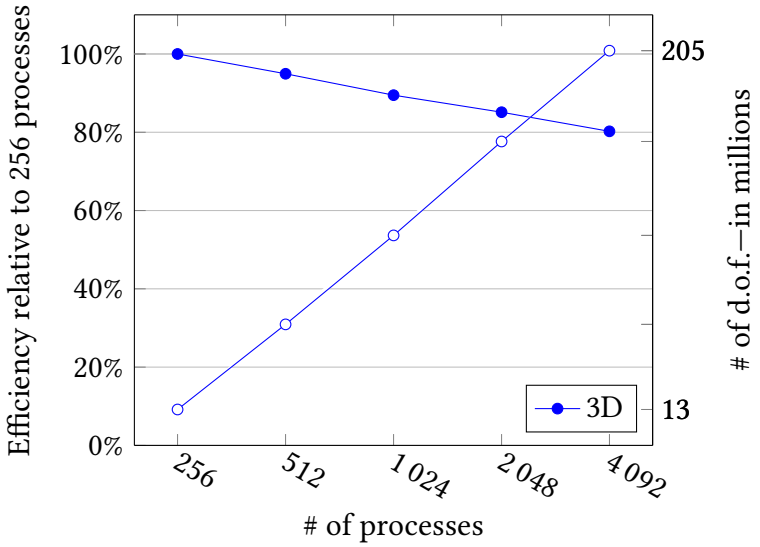


Fig 5.2 – Weak scaling experiments in Navier-Stokes system for a full time step timings tab .

	N	Factorization	GCRODR	# of it.	Total	# of d.o.f.
3D	256	28.2 s	17.0 s	29	67.9 s	$13.7 \cdot 10^6$
	512	30.5 s	16.7 s	28	68.1 s	$26.0 \cdot 10^6$
	1 024	30.6 s	19.7 s	35	70.8 s	$50.9 \cdot 10^6$
	2 048	30.9 s	24.0 s	41	75.7 s	$103.6 \cdot 10^6$
	4 092	28.6 s	28.5 s	57	79.8 s	$205.8 \cdot 10^6$

Fig 5.3 – Weak scaling experiments in Navier-Stokes system for a full time step timings tab .

Weak scalability These computations, see Figure 5.3 and Figure 5.2, assess the weak scalability of the algorithm with respect to the problem size and the number of subdomains. All times are wall clock times.
The domain is decomposed automatically into subdomains with a graph partitioner, ranging from 256 to 4092 subdomains. The problem size is increased by mesh refinement. In our 3D Turek Benchmark the initial problem is about 13 millions d.o.f decomposed into 256 subdomains and solved in 67.9s and the final problem is about 205 millions of d.o.f decomposed into 4092 subdomains and solved in 79.8s which gives an efficiency near to 80%.
In figure table 5.3, we report the number of GCRO-DR iterations. Thanks to the robustness of the domain decomposition preconditioner with the recycling of the Krylov subspace, we can observe that they algorithm remain quite stable in terms of number of iterations . We report in the same table all the timings concerning the algorithm, column "Factorization" concerns the local subdomains, in column "GCRO-DR" we display the time spent by Krylov solver GCRO-DR.

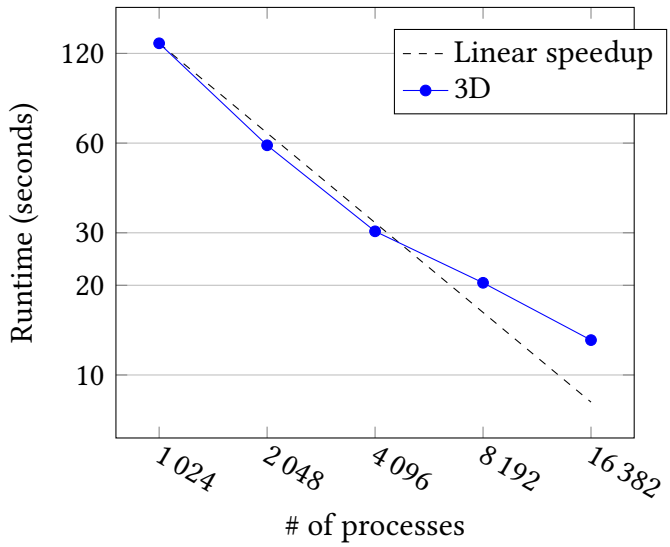


Fig 5.4 – Strong scaling experiments in Navier-Stokes system for a full time step timings curve.

	N	Factorization	GCRODR	# of it.	Total Step	# of d.o.f.
	1 024	70.4 s	27.2 s	36	129.7 s	
	2 048	23.7 s	16.7 s	46	59.0 s	
3D	4 096	8.1 s	10.3 s	57	30.3 s	$87.11 \cdot 10^6$
	8 192	3.6 s	7.0 s	66	20.4 s	
	16 384	1.5 s	4.7 s	75	13.1 s	

Fig 5.5 – Strong scaling experiments in Navier-Stokes system for a full entire time step timings tab .

Strong scalability We assess here the strong scalability of the ORAS (5.7) preconditioner equipped with the recycling technics GCRO-DR on the Turek benchmark. For this, we make the number of subdomains vary for a fixed global system size. In our test case the system size is fixed to 87 millions unknowns (d.o.f). All times are wall clock times. The domain is decomposed automatically into subdomains with a graph partition, ranging from 1024 to 16384 subdomains. In 3D as we can show in figure 5.4, from 1024 subdomains to 16384 subdomains we get a quite good speed up. We observe a linear speedup until 4092 processors, then we lose a little bit the linear speed up behavior but the results are still reasonable.
Throughout the simulations on a varying number of processors, for performing an entire time step in the simulation, we pass from 128.7s using 1024 subdomains to 13.1s when using 16384 subdomains. In figure table 5.5 we display all timings relative to this test. Column "Factorization" gives the time spent in the factorization of the local submatrices, column "GCRO-DR" is the time taken by the GCRO-DR solve of the global linear system by the domain decomposition algorithm.

Chapter 6

Conclusion

Contents

Throughout this thesis we have developed an improvement for Schwarz domain decomposition methods, first by incorporating more sophisticated transmission condition which allows us to reach more optimal convergence, this was possible by the use of Robin conditions, then we developed a mathematical framework, where we equipped the optimised Schwarz method with an automatic optimized coarse space, this lead to a two level method called "SORAS-GenEO-2" with a guaranteed convergence rates. Numerical extension was done to tackle incompressible PDEs systems such as Stokes system, then for unstationnary Navier-Stokes systems. Thanks to the robustness of the constructed methods and the parallel framework given by HPDDM code, we were able to conduct Large-Scale parallel simulation for various complex mechanical problems.

Of course there are still open questions and a very promising research directions.

6.1 Possible future work & open questions

- The two level method SORAS-GEnEO-2 is based on solving local eigenvalue problems, this allows a very high robustness when dealing with large-scale simulations, however the time spent in solving these eigenvalues problems can not be neglectable, and becomes more and more important which may produce botlenecks. This lead us to explore some direction of research in order to handle this question.
  - One of the direction we can explore is how to generalize the SORAS-GEnEO-2 method to a multilevel method when the dimension of the coarse space becomes important. As it is the case for BDDC methods.
- Simulation that couple multiple physical phenomena is becoming a general practice in domain of scientific prediction. Multiphysics simulations [Key+13] need a computational environment that permit the extreme scale to handle the challenging simulations. Domain decomposition methods are still problem dependent, among the unresolved questions, one question consists in how to design domain decomposition methods that handle multiphisics nature of the problems. One of the future work will consist in observing

and improving the behavior of the developed method on the monolithic coupling of fluid-structure interaction problems.

Appendix A

Appendix: Maple computations

Contents

In this section we explain more deeply the sequence of Maple computations, after the application of the Fourier transform to the 2D elasticity system, we get the following ODE's system

```
> ode1 := (2*mu+lambda)*(diff(u(x),[x$2]))-mu*k*k*u(x)+I*k*(mu+lambda)*(diff(v(x),x));
> ode2 := mu*(diff(v(x),[x$2]))-(2*mu+lambda)*k*k*v(x)+I*k*(mu+lambda)*(diff(u(x),x));
```

$$ode1 := (2\,\mu+\lambda)\,\frac{\mathrm{d}^2}{\mathrm{d}x^2}u(x) - \mu\,k^2 u(x) + ik\,(\mu+\lambda)\,\frac{\mathrm{d}}{\mathrm{d}x}v(x)$$

$$ode2 := \mu\,\frac{\mathrm{d}^2}{\mathrm{d}x^2}v(x) - (2\,\mu+\lambda)\,k^2 v(x) + ik\,(\mu+\lambda)\,\frac{\mathrm{d}}{\mathrm{d}x}u(x)$$

dsolve function of Maple give the following solutions

```
> dsol := dsolve(ode1, ode2);
```

$$dsol := u(x) = \_C1\,\mathrm{e}^{kx} + \_C2\,\mathrm{e}^{kx}x + \_C3\,\mathrm{e}^{-kx} + \_C4\,\mathrm{e}^{-kx}x$$

$$v(x) = \frac{-i\left(\mathrm{e}^{-kx}\_C4\,k\lambda\,x + \mathrm{e}^{-kx}\_C4\,k\mu\,x - \mathrm{e}^{kx}\_C2\,k\lambda\,x - \mathrm{e}^{kx}\_C2\,k\mu\,x + \mathrm{e}^{-kx}\_C3\,k\lambda\right)}{k\,(\mu+\lambda)}$$

$$\frac{-i\left(\mathrm{e}^{-kx}\_C3\,k\mu - \mathrm{e}^{kx}\_C1\,k\lambda - \mathrm{e}^{kx}\_C1\,k\mu - \mathrm{e}^{-kx}\_C4\,\lambda - 3\,\mathrm{e}^{-kx}\_C4\,\mu\right)}{k\,(\mu+\lambda)}$$

$$\frac{-i\left(-\mathrm{e}^{kx}\_C2\,\lambda - 3\,\mathrm{e}^{kx}\_C2\,\mu\right)}{k\,(\mu+\lambda)}$$

The solution must be bounded at infinity so we can reduce the expressions in each subdomains (Ω_1,Ω_2)

```
> u1 := coeff(rhs(dsol[1]), exp(k*x))*exp(k*x);
> v1 := coeff(rhs(dsol[2]), exp(k*x))*exp(k*x);
> u2 := coeff(rhs(dsol[1]), exp(-k*x))*exp(-k*x);
> v2 := coeff(rhs(dsol[2]), exp(-k*x))*exp(-k*x);
```

$$u1 := (\_C2\,x + \_C1)\,\mathrm{e}^{kx}$$
$$v1 := \frac{-i\,(-\_C2\,k\lambda\,x - \_C2\,k\mu\,x - \_C1\,k\lambda - \_C1\,k\mu - \_C2\,\lambda - 3\,\_C2\,\mu)\,\mathrm{e}^{kx}}{k\,(\mu+\lambda)}$$
$$u2 := (\_C4\,x + \_C3)\,\mathrm{e}^{-kx}$$
$$v2 := \frac{-i\,(\_C4\,k\lambda\,x + \_C4\,k\mu\,x + \_C3\,k\lambda + \_C3\,k\mu - \_C4\,\lambda - 3\,\_C4\,\mu)\,\mathrm{e}^{-kx}}{k\,(\mu+\lambda)}$$

By the following we homogenize the notations of the constant integration in the two subdomain.

```
> u1 := eval(eval(eval(eval(u1, _C1 = a1), _C3 = a1), _C2 = b1), _C4 = b1);
> v1 := eval(eval(eval(eval(v1, _C1 = a1), _C3 = a1), _C2 = b1), _C4 = b1);
> u2 := eval(eval(eval(eval(u2, _C1 = a2), _C3 = a2), _C2 = b2), _C4 = b2);
> v2 := eval(eval(eval(eval(v2, _C1 = a2), _C3 = a2), _C2 = b2), _C4 = b2);
```

$$u1 := (b1\,x + a1)\,\mathrm{e}^{kx}$$
$$v1 := \frac{-i\,(-b1\,k\lambda\,x - b1\,k\mu\,x - a1\,k\lambda - a1\,k\mu - b1\,\lambda - 3\,b1\,\mu)\,\mathrm{e}^{kx}}{k\,(\mu+\lambda)}$$
$$u2 := (b2\,x + a2)\,\mathrm{e}^{-kx}$$
$$v2 := \frac{-i\,(b2\,k\lambda\,x + b2\,k\mu\,x + a2\,k\lambda + a2\,k\mu - b2\,\lambda - 3\,b2\,\mu)\,\mathrm{e}^{-kx}}{k\,(\mu+\lambda)}$$

A.1 How we compute the DtN operator ?

In this section we are going to explain how we construct the DtN operator algebraically using the matrix dependency of the local solution on the undetermined constants.

compute DtN_2**?** *For this we use interface condition*

$$\left(\underline{\underline{\sigma}}_S.n_1 + DtN_2\right)\begin{bmatrix} u_1^{n+1} \\ v_1^{n+1}\end{bmatrix}(x,k) = \left(\underline{\underline{\sigma}}_S.n_1 + DtN_2\right)\begin{bmatrix} u_2^{n} \\ v_2^{n}\end{bmatrix}(x,k)$$

We construct the Operator DtN_2 *such that*

$$\left(\underline{\underline{\sigma}}_S.n_1 + DtN_2\right)\begin{bmatrix} u_2^{n} \\ v_2^{n}\end{bmatrix}(x,k) = 0.$$

For the subdomain Ω_2, $[u_2, v_2]^\intercal$ is expressed according to a_2, b_2, in the following we deduce the matrix form of both $[u_2, v_2]^\intercal$ and $\underline{\underline{\sigma}}_S([u_2, v_2]^\intercal).n$

$$\begin{bmatrix} u2 \\ v2 \end{bmatrix} = A_2 \begin{bmatrix} a2 \\ b2 \end{bmatrix}$$

```
> eqA := eval(u2, x = x);
> eqB := eval(v2, x = x);
```

$$eqA := (b2\,x + a2)\,\mathrm{e}^{-kx}$$

$$eqB := \frac{-i\,(b2\,k\lambda\,x + b2\,k\mu\,x + a2\,k\lambda + a2\,k\mu - b2\,\lambda - 3\,b2\,\mu)\,\mathrm{e}^{-kx}}{k\,(\mu + \lambda)}$$

```
> A2 := simplify(matrix([[coeff(eqA, a2), coeff(eqA, b2)], [coeff(eqB, a2), coeff(eqB, b2)]]));
```

$$A2 := \begin{bmatrix} \mathrm{e}^{-kx} & x\mathrm{e}^{-kx} \\ -i\mathrm{e}^{-kx} & \dfrac{-i\,(k\lambda\,x + k\mu\,x - \lambda - 3\,\mu)\,\mathrm{e}^{-kx}}{k\,(\mu + \lambda)} \end{bmatrix}$$

Then we can also expresse according to a_2, b_2 the stress Cauchy tensor in the normal direction $\underline{\underline{\sigma}}_S([u_2, v_2]^\intercal).n$ as

$$\underline{\underline{\sigma}}_S([u_2, v_2]^\intercal).n = \begin{bmatrix} \underline{\underline{\sigma}}_S^N \\ \underline{\underline{\sigma}}_S^t \end{bmatrix} = \begin{bmatrix} \lambda\left(\dfrac{\partial u_2}{\partial x} + ik\dfrac{\partial v_2}{\partial y}\right) + 2\mu\dfrac{\partial u_2}{\partial x} \\ \mu\left(\dfrac{\partial v_2}{\partial x} + ik\dfrac{\partial u_2}{\partial y}\right) \end{bmatrix} = B2 \begin{bmatrix} a2 \\ b2 \end{bmatrix}$$

```
> eq1 := eval(-(2*mu+lambda)*(diff(u2, x))-I*lambda*k*v2, x = x); 1; eq2 := eval(-mu*(diff(v2, x))-I*mu*k*u2, x = x);
```

$$eq1 := -(2\,\mu + \lambda)\left(b2\,\mathrm{e}^{-kx} - (b2\,x + a2)\,k\mathrm{e}^{-kx}\right) + \frac{\lambda\,(-b2\,k\lambda\,x - b2\,k\mu\,x - a2\,k\lambda - a2\,k\mu + b2\,\lambda + 3\,b2\,\mu)\,\mathrm{e}^{-kx}}{\mu + \lambda}$$

$$eq2 := -\mu\left(\frac{i\,(-b2\,k\lambda - b2\,k\mu)\,\mathrm{e}^{-kx}}{k\,(\mu + \lambda)} - \frac{i\,(-b2\,k\lambda\,x - b2\,k\mu\,x - a2\,k\lambda - a2\,k\mu + b2\,\lambda + 3\,b2\,\mu)\,\mathrm{e}^{-kx}}{\mu + \lambda}\right) - I * \mu * k * (b2 * x + a2) * exp(-k * x) - i\mu\,k\,(b2\,x + a2)\,\mathrm{e}^{-kx}$$

```
> B2 := simplify(matrix([[coeff(eq1, a2), coeff(eq1, b2)], [coeff(eq2, a2), coeff(eq2, b2)]])); 1
```

$$B2 := \begin{bmatrix} 2\,\mu\,k\mathrm{e}^{-kx} & 2\,\dfrac{\mathrm{e}^{-kx}\mu\,(k\lambda\,x+k\mu\,x-\mu)}{\mu+\lambda} \\ -2\,i\mu\,\mathrm{e}^{-kx}k & \dfrac{-2\,i\mu\,\mathrm{e}^{-kx}\,(k\lambda\,x+k\mu\,x-\lambda-2\,\mu)}{\mu+\lambda} \end{bmatrix}$$

So,

$$\left(\underline{\underline{\sigma}}_S.n_1 + DtN_2\right)\begin{bmatrix} u_2^n \\ v_2^n \end{bmatrix}(x,k) = 0. \Leftrightarrow B2\begin{bmatrix} a2 \\ b2 \end{bmatrix} + DtN2A2\begin{bmatrix} a2 \\ b2 \end{bmatrix} = 0.$$

We can deduce now the purely algebraic operator $DtN2$ by

```
> DtN2 := simplify(evalm((B2 &* inverse(A2))));
```

$$DtN2 := \begin{bmatrix} 2\,\dfrac{k\mu\,(2\,\mu+\lambda)}{\lambda+3\,\mu} & \dfrac{2\,i\mu^2k}{\lambda+3\,\mu} \\ \dfrac{-2\,ik\mu^2}{\lambda+3\,\mu} & 2\,\dfrac{k\mu\,(2\,\mu+\lambda)}{\lambda+3\,\mu} \end{bmatrix}$$

```
> DtN2 := eval(DtN2, k = kzero);
```

$$DtN2 := \begin{bmatrix} 2\,\dfrac{kzero\,\mu\,(2\,\mu+\lambda)}{\lambda+3\,\mu} & \dfrac{2\,i\mu^2kzero}{\lambda+3\,\mu} \\ \dfrac{-2\,i\mu^2kzero}{\lambda+3\,\mu} & 2\,\dfrac{kzero\,\mu\,(2\,\mu+\lambda)}{\lambda+3\,\mu} \end{bmatrix}$$

Compute DtN_1? *We conduct the same method as it was used to determine DtN_2, the only difference lies on the fact of using the seconde interface condition*

$$\left(\underline{\underline{\sigma}}_S.n_1 + DtN_2\right)\begin{bmatrix} u_2^{n+1} \\ v_2^{n+1} \end{bmatrix}(x,k) = \left(\underline{\underline{\sigma}}_S.n_2 + DtN_1\right)\begin{bmatrix} u_1^n \\ v_1^n \end{bmatrix}(x,k)$$

We construct the Operator DtN_1 such that

$$\left(\underline{\underline{\sigma}}_S.n_2 + DtN_1\right)\begin{bmatrix} u_1^n \\ v_1^n \end{bmatrix}(x,k) = 0.$$

$$\begin{bmatrix} u1 \\ v1 \end{bmatrix} = A_1\begin{bmatrix} a1 \\ b1 \end{bmatrix}$$

```
> eqC := eval(u1, x = x);
> eqD := eval(v1, x = x);
```

$$eqC := (b1\,x + a1)\,\mathrm{e}^{kx}$$

$$eqD := \frac{i\,(b1\,k\lambda\,x + b1\,k\mu\,x + a1\,k\lambda + a1\,k\mu + b1\,\lambda + 3\,b1\,\mu)\,\mathrm{e}^{kx}}{k\,(\mu+\lambda)}$$

```
> A1 := simplify(matrix([[coeff(eqC, a1), coeff(eqC, b1)], [coeff(eqD, a1), coeff(eqD, b1)]]));
```

$$A1 := \begin{bmatrix} e^{kx} & x e^{kx} \\ i e^{kx} & \dfrac{i\,(k\lambda\,x + k\mu\,x + \lambda + 3\,\mu)\,e^{kx}}{k\,(\mu+\lambda)} \end{bmatrix}$$

The seconde matrix form for $\underline{\underline{\sigma}}_S([u_1, v_1]^\intercal).n$

$$\underline{\underline{\sigma}}_S([u_1, v_1]^\intercal).n = \begin{bmatrix} \underline{\sigma}_S^N \\ \underline{\sigma}_S^t \end{bmatrix} = \begin{bmatrix} \lambda\left(\dfrac{\partial u_1}{\partial x} + ik\dfrac{\partial v_1}{\partial y}\right) + 2\mu\dfrac{\partial u_1}{\partial x} \\ \mu\left(\dfrac{\partial v_1}{\partial x} + ik\dfrac{\partial u_1}{\partial y}\right) \end{bmatrix} = B1 \begin{bmatrix} a1 \\ b1 \end{bmatrix}$$

```
> eq3 := eval((2*mu+lambda)*(diff(u1, x))+I*lambda*k*v1, x = x);
> eq4 := eval(mu*(diff(v1, x))+I*mu*k*u1, x = x)
```

$$eq3 := (2\,\mu+\lambda)\left(b1\,e^{kx} + (b1\,x + a1)\,k e^{kx}\right) - \frac{\lambda\,(b1\,k\lambda\,x + b1\,k\mu\,x + a1\,k\lambda + a1\,k\mu + b1\,\lambda + 3\,b1\,\mu)\,e^{kx}}{\mu+\lambda}$$

$$eq4 := \mu\left(\frac{i\,(b1\,k\lambda + b1\,k\mu)\,e^{kx}}{k\,(\mu+\lambda)} + \frac{i\,(b1\,k\lambda\,x + b1\,k\mu\,x + a1\,k\lambda + a1\,k\mu + b1\,\lambda + 3\,b1\,\mu)\,e^{kx}}{\mu+\lambda}\right) + i\mu\,k\,(b1\,x + a1)\,e^{kx}$$

```
> B1 := simplify(matrix([[coeff(eq3, a1), coeff(eq3, b1)], [coeff(eq4, a1), coeff(eq4, b1)]]));
```

$$B1 := \begin{bmatrix} 2\,\mu\,k e^{kx} & 2\,\dfrac{e^{kx}\mu\,(k\lambda\,x + k\mu\,x + \mu)}{\mu+\lambda} \\ 2\,i\mu\,k e^{kx} & \dfrac{2\,i\mu\,e^{kx}\,(k\lambda\,x + k\mu\,x + \lambda + 2\,\mu)}{\mu+\lambda} \end{bmatrix}$$

hence

$$\left(\underline{\underline{\sigma}}_S.n_1 + DtN_1\right)\begin{bmatrix} u_1^n \\ v_1^n \end{bmatrix}(x,k) = 0. \Leftrightarrow B1\begin{bmatrix} a1 \\ b1 \end{bmatrix} + DtN_1 A1\begin{bmatrix} a1 \\ b1 \end{bmatrix} = 0.$$

```
> DtN1 := simplify(evalm((B2 & *inverse(A2))));
```

$$DtN1 := \begin{bmatrix} 2\,\dfrac{k\mu\,(2\,\mu+\lambda)}{\lambda+3\,\mu} & \dfrac{-2\,ik\mu^2}{\lambda+3\,\mu} \\ \dfrac{2\,i\mu^2 k}{\lambda+3\,\mu} & 2\,\dfrac{k\mu\,(2\,\mu+\lambda)}{\lambda+3\,\mu} \end{bmatrix}$$

```
> DtN1 := eval(DtN1, k = kzero);
```

$$DtN1 := \begin{bmatrix} 2\,\dfrac{kzero\,\mu\,(2\,\mu+\lambda)}{\lambda+3\,\mu} & \dfrac{-2\,i\mu^2 kzero}{\lambda+3\,\mu} \\ \dfrac{2\,i\mu^2 kzero}{\lambda+3\,\mu} & 2\,\dfrac{kzero\,\mu\,(2\,\mu+\lambda)}{\lambda+3\,\mu} \end{bmatrix}$$

A.2 Convergence rate

if we summarize what we have done in the previous sections, we have expressed the error solutions of the algorithm in each subdomain according to the integration constants a_1, b_1, a_2, b_2. in order to determine these constants, boundary conditions Robin boundary conditions are considered, but before, we have determined DtN1, and DtN2, wich will be used inside a Robin conditions for Schwarz algorithm.

Now we study the convergence behavior of the Schwarz algorithm by describing the behavior of the integration constants a_1, b_1, a_2, b_2 via the iteration matrices.

$$\begin{bmatrix} a_1 \\ b_1 \end{bmatrix}^{n+1} = M_1 \begin{bmatrix} a_1 \\ b_1 \end{bmatrix}^{n} \text{ and } \begin{bmatrix} a_2 \\ b_2 \end{bmatrix}^{n+1} = M_2 \begin{bmatrix} a_2 \\ b_2 \end{bmatrix}^{n}$$

To determine M_1 and M_2 we use the following Robin conditions

$$\begin{cases} \left[\underline{\underline{\sigma}}_S.n_1 + DtN2\right] \begin{bmatrix} u_1^{n+1} \\ v_1^{n+1} \end{bmatrix} (\delta, y) = \left[\underline{\underline{\sigma}}_S.n_1 + DtN2\right] \begin{bmatrix} u_2^{n} \\ v_2^{n} \end{bmatrix} (\delta, y),\ y \in \mathbb{R}, \\ \left[\underline{\underline{\sigma}}_S.n_2 + DtN1\right] \begin{bmatrix} u_2^{n+1} \\ v_2^{n+1} \end{bmatrix} (0, y) = \left[\underline{\underline{\sigma}}_S.n_2 + DtN1\right] \begin{bmatrix} u_1^{n} \\ v_1^{n} \end{bmatrix} (0, y),\ y \in \mathbb{R}, \end{cases}$$

we recall that the solution in each subdomain reads

$$\begin{aligned} u1 &:= (b1\,x + a1)\,\mathrm{e}^{kx} \\ v1 &:= \frac{-i\,(-b1\,k\lambda\,x - b1\,k\mu\,x - a1\,k\lambda - a1\,k\mu - b1\,\lambda - 3\,b1\,\mu)\,\mathrm{e}^{kx}}{k\,(\mu+\lambda)} \\ u2 &:= (b2\,x + a2)\,\mathrm{e}^{-kx} \\ v2 &:= \frac{-i\,(b2\,k\lambda\,x + b2\,k\mu\,x + a2\,k\lambda + a2\,k\mu - b2\,\lambda - 3\,b2\,\mu)\,\mathrm{e}^{-kx}}{k\,(\mu+\lambda)} \end{aligned}$$

From

$$\left[\underline{\underline{\sigma}}_S.n_1 + DtN2\right] \begin{bmatrix} u_1^{n+1} \\ v_1^{n+1} \end{bmatrix} (\delta, y) = \left[\underline{\underline{\sigma}}_S.n_1 + DtN2\right] \begin{bmatrix} u_2^{n} \\ v_2^{n} \end{bmatrix} (\delta, y)$$

we get

$$Ad_1 \begin{bmatrix} a_1 \\ b_1 \end{bmatrix}^{n+1} = Ad_2 \begin{bmatrix} a_2 \\ b_2 \end{bmatrix}^{n}$$

```
> eq5 := eval((2*mu+lambda)*(diff(u1, x))+I*lambda*k*v1+DtN2[1, 1]*u1+DtN2[1,
2]*v1-(2*mu+lambda)*(diff(u2, x))-I*lambda*k*v2-DtN2[1, 1]*u2-DtN2[1, 2]*v2, x =
L1);
```

$$
\begin{aligned}
eq5 :=\ & (2\,\mu+\lambda)\left(b1\,\mathrm{e}^{kL1}+(L1\,b1+a1)\,k\mathrm{e}^{kL1}\right)\\
&+\frac{\lambda\,(-L1\,b1\,k\lambda-L1\,b1\,k\mu-a1\,k\lambda-a1\,k\mu-b1\,\lambda-3\,b1\,\mu)\,\mathrm{e}^{kL1}}{\mu+\lambda}\\
&+2\,\frac{kzero\,\mu\,(2\,\mu+\lambda)\,(L1\,b1+a1)\,\mathrm{e}^{kL1}}{\lambda+3\,\mu}\\
&+2\,\frac{\mu^2 kzero\,(-L1\,b1\,k\lambda-L1\,b1\,k\mu-a1\,k\lambda-a1\,k\mu-b1\,\lambda-3\,b1\,\mu)\,\mathrm{e}^{kL1}}{(\lambda+3\,\mu)\,k\,(\mu+\lambda)}\\
&-(2\,\mu+\lambda)\left(b2\,\mathrm{e}^{-kL1}-(L1\,b2+a2)\,k\mathrm{e}^{-kL1}\right)\\
&-\frac{\lambda\,(L1\,b2\,k\lambda+L1\,b2\,k\mu+a2\,k\lambda+a2\,k\mu-b2\,\lambda-3\,b2\,\mu)\,\mathrm{e}^{-kL1}}{\mu+\lambda}\\
&-2\,\frac{kzero\,\mu\,(2\,\mu+\lambda)\,(L1\,b2+a2)\,\mathrm{e}^{-kL1}}{\lambda+3\,\mu}\\
&-2\,\frac{\mu^2 kzero\,(L1\,b2\,k\lambda+L1\,b2\,k\mu+a2\,k\lambda+a2\,k\mu-b2\,\lambda-3\,b2\,\mu)\,\mathrm{e}^{-kL1}}{(\lambda+3\,\mu)\,k\,(\mu+\lambda)}
\end{aligned}
$$

```
> eq6 := eval(mu*(diff(v1, x))+I*mu*k*u1+DtN2[2, 1]*u1+DtN2[2, 2]*v1-mu*(diff(v2,
x))-I*mu*k*u2-DtN2[2, 1]*u2-DtN2[2, 2]*v2, x = L1);
```

$$
\begin{aligned}
eq6 :=\ & \mu\left(\frac{-i\,(-b1\,k\lambda-b1\,k\mu)\,\mathrm{e}^{kL1}}{k\,(\mu+\lambda)}-\frac{i\,(-L1\,b1\,k\lambda-L1\,b1\,k\mu-a1\,k\lambda-a1\,k\mu-b1\,\lambda-3\,b1\,\mu)\,\mathrm{e}^{kL1}}{\mu+\lambda}\right)\\
&+i\mu\,k\,(L1\,b1+a1)\,\mathrm{e}^{kL1}-\frac{2\,i\mu^2 kzero\,(L1\,b1+a1)\,\mathrm{e}^{kL1}}{\lambda+3\,\mu}\\
&-\frac{2\,ikzero\,\mu\,(2\,\mu+\lambda)\,(-L1\,b1\,k\lambda-L1\,b1\,k\mu-a1\,k\lambda-a1\,k\mu-b1\,\lambda-3\,b1\,\mu)\,\mathrm{e}^{kL1}}{(\lambda+3\,\mu)\,k\,(\mu+\lambda)}\\
&-\mu\left(\frac{-i\,(b2\,k\lambda+b2\,k\mu)\,\mathrm{e}^{-kL1}}{k\,(\mu+\lambda)}+\frac{i\,(L1\,b2\,k\lambda+L1\,b2\,k\mu+a2\,k\lambda+a2\,k\mu-b2\,\lambda-3\,b2\,\mu)\,\mathrm{e}^{-kL1}}{\mu+\lambda}\right)\\
&-i\mu\,k\,(L1\,b2+a2)\,\mathrm{e}^{-kL1}+\frac{2\,i\mu^2 kzero\,(L1\,b2+a2)\,\mathrm{e}^{-kL1}}{\lambda+3\,\mu}\\
&+\frac{2\,ikzero\,\mu\,(2\,\mu+\lambda)\,(L1\,b2\,k\lambda+L1\,b2\,k\mu+a2\,k\lambda+a2\,k\mu-b2\,\lambda-3\,b2\,\mu)\,\mathrm{e}^{-kL1}}{(\lambda+3\,\mu)\,k\,(\mu+\lambda)}
\end{aligned}
$$

From

$$\left[\underline{\underline{\sigma}}_S.n_2+DtN1\right]\begin{bmatrix}u_2^{n+1}\\ v_2^{n+1}\end{bmatrix}(0,y)=\left[\underline{\underline{\sigma}}_S.n_2+DtN1\right]\begin{bmatrix}u_1^{n}\\ v_1^{n}\end{bmatrix}(0,y),\ y\in\mathbb{R},$$

we get

$$Bd_2\begin{bmatrix}a_2\\ b_2\end{bmatrix}^{n+1}=Bd_1\begin{bmatrix}a_1\\ b_1\end{bmatrix}^{n}$$

```
> eq7 := eval(-(2*mu+lambda)*(diff(u2, x))-I*lambda*k*v2+DtN1[1, 1]*u2+DtN2[1,
2]*v2+((2*mu+lambda)*(diff(u1, x))+I*lambda*k*v1-DtN1[1, 1]*u1-DtN2[1, 2]*v1), x
= L2);
```

$$
\begin{aligned}
eq7 :=\ & -(2\mu+\lambda)\left(b2\,e^{-kL2}-(L2\,b2+a2)\,k e^{-kL2}\right)\\
& -\frac{\lambda\,(L2\,b2\,k\lambda+L2\,b2\,k\mu+a2\,k\lambda+a2\,k\mu-b2\,\lambda-3\,b2\,\mu)\,e^{-kL2}}{\mu+\lambda}\\
& +2\,\frac{kzero\,\mu\,(2\mu+\lambda)\,(L2\,b2+a2)\,e^{-kL2}}{\lambda+3\mu}\\
& +2\,\frac{\mu^2 kzero\,(L2\,b2\,k\lambda+L2\,b2\,k\mu+a2\,k\lambda+a2\,k\mu-b2\,\lambda-3\,b2\,\mu)\,e^{-kL2}}{(\lambda+3\mu)\,k\,(\mu+\lambda)}\\
& +(2\mu+\lambda)\left(b1\,e^{kL2}+(L2\,b1+a1)\,k e^{kL2}\right)\\
& +\frac{\lambda\,(-L2\,b1\,k\lambda-L2\,b1\,k\mu-a1\,k\lambda-a1\,k\mu-b1\,\lambda-3\,b1\,\mu)\,e^{kL2}}{\mu+\lambda}\\
& -2\,\frac{kzero\,\mu\,(2\mu+\lambda)\,(L2\,b1+a1)\,e^{kL2}}{\lambda+3\mu}\\
& -2\,\frac{\mu^2 kzero\,(-L2\,b1\,k\lambda-L2\,b1\,k\mu-a1\,k\lambda-a1\,k\mu-b1\,\lambda-3\,b1\,\mu)\,e^{kL2}}{(\lambda+3\mu)\,k\,(\mu+\lambda)}
\end{aligned}
$$

```
> eq8 := eval(-mu*(diff(v2, x))-I*mu*k*u2+DtN1[2, 1]*u2+DtN1[2, 2]*v2+(mu*(diff(v1,
x))+I*mu*k*u1-DtN1[2, 1]*u1-DtN1[2, 2]*v1), x = L2);
```

$$
\begin{aligned}
eq8 :=\ & -\mu\left(\frac{-i\,(b2\,k\lambda+b2\,k\mu)\,e^{-kL2}}{k\,(\mu+\lambda)}+\frac{i\,(L2\,b2\,k\lambda+L2\,b2\,k\mu+a2\,k\lambda+a2\,k\mu-b2\,\lambda-3\,b2\,\mu)\,e^{-kL2}}{\mu+\lambda}\right)\\
& -i\mu\,k\,(L2\,b2+a2)\,e^{-kL2}+\frac{2\,i\mu^2 kzero\,(L2\,b2+a2)\,e^{-kL2}}{\lambda+3\mu}\\
& -\frac{2\,ikzero\,\mu\,(2\mu+\lambda)\,(L2\,b2\,k\lambda+L2\,b2\,k\mu+a2\,k\lambda+a2\,k\mu-b2\,\lambda-3\,b2\,\mu)\,e^{-kL2}}{(\lambda+3\mu)\,k\,(\mu+\lambda)}\\
& +\mu\left(\frac{-i\,(-b1\,k\lambda-b1\,k\mu)\,e^{kL2}}{k\,(\mu+\lambda)}-\frac{i\,(-L2\,b1\,k\lambda-L2\,b1\,k\mu-a1\,k\lambda-a1\,k\mu-b1\,\lambda-3\,b1\,\mu)\,e^{kL2}}{\mu+\lambda}\right)\\
& +i\mu\,k\,(L2\,b1+a1)\,e^{kL2}\\
& -\frac{2\,i\mu^2 kzero\,(L2\,b1+a1)\,e^{kL2}}{\lambda+3\mu}\\
& +\frac{2\,ikzero\,\mu\,(2\mu+\lambda)\,(-L2\,b1\,k\lambda-L2\,b1\,k\mu-a1\,k\lambda-a1\,k\mu-b1\,\lambda-3\,b1\,\mu)\,e^{kL2}}{(\lambda+3\mu)\,k\,(\mu+\lambda)}
\end{aligned}
$$

```
> Ad1 := simplify(matrix([[coeff(eq5, a1), coeff(eq5, b1)], [coeff(eq6, a1), coeff(eq6,
b1)]]));
```

$$Ad1 := \begin{bmatrix} \frac{(2\,k\lambda + 6\,k\mu + 2\,kzero\,\lambda + 2\,kzero\,\mu)\,\mu\,e^{kL1}}{\lambda + 3\,\mu} & 2\,\frac{e^{kL1}\mu\,\left(L1\,k^2\lambda^2 + 4\,L1\,k^2\lambda\,\mu + 3\,L1\,k^2\mu^2 + L1\,kkzero\,\lambda^2 + 2\,kzero\,\mu\,L1\,k\lambda + kzero\,\mu^2 L1\,k + k\lambda\,\mu + 3\,k\mu^2 - \right.}{(\lambda + 3\,\mu)\,k\,(\mu + \lambda)} \\ \frac{2\,i\,(k\lambda + 3\,k\mu + kzero\,\lambda + kzero\,\mu)\,\mu\,e^{kL1}}{\lambda + 3\,\mu} & \frac{2\,i\mu\,e^{kL1}\left(L1\,k^2\lambda^2 + 4\,L1\,k^2\lambda\,\mu + 3\,L1\,k^2\mu^2 + L1\,kkzero\,\lambda^2 + 2\,kzero\,\mu\,L1\,k\lambda + kzero\,\mu^2 L1\,k + k\lambda^2 + 5\,k\lambda\,\mu + 6\,k\mu^2 + k\right.}{(\lambda + 3\,\mu)\,k\,(\mu + \lambda)} \end{bmatrix}$$

```
> Ad2 := simplify(matrix([[coeff(eq5, a2), coeff(eq5, b2)], [coeff(eq6, a2), coeff(eq6, b2)]]));
```

$$Ad2 := \begin{bmatrix} 2\,e^{-kL1}\mu\,(k - kzero) & \frac{\left(2\,L1\,k^2\lambda + 2\,L1\,k^2\mu - 2\,L1\,kkzero\,\lambda - 2\,kzero\,\mu\,L1\,k - 2\,k\mu + 2\,kzero\,\mu\right)\,\mu\,e^{-kL1}}{k\,(\mu + \lambda)} \\ -2\,ie^{-kL1}\mu\,(k - kzero) & \frac{-2\,i\left(L1\,k^2\lambda + L1\,k^2\mu - L1\,kkzero\,\lambda - kzero\,\mu\,L1\,k - k\lambda - 2\,k\mu + kzero\,\lambda + 2\,kzero\,\mu\right)\,\mu\,e^{-kL1}}{k\,(\mu + \lambda)} \end{bmatrix}$$

```
> Bd1 := simplify(matrix([[coeff(eq7, a1), coeff(eq7, b1)], [coeff(eq8, a1), coeff(eq8, b1)]]));
```

$$Bd1 := \begin{bmatrix} \frac{(2\,k\lambda + 6\,k\mu - 2\,kzero\,\lambda - 2\,kzero\,\mu)\,\mu\,e^{kL2}}{\lambda + 3\,\mu} & 2\,\frac{e^{kL2}\mu\,\left(L2\,k^2\lambda^2 + 4\,L2\,k^2\lambda\,\mu + 3\,L2\,k^2\mu^2 - L2\,kkzero\,\lambda^2 - 2\,kzero\,\mu\,L2\,k\lambda - kzero\,\mu^2 L2\,k + k\lambda\,\mu + 3\,k\mu^2 + kzero\,\mu\,\lambda + \right.}{(\lambda + 3\,\mu)\,k\,(\mu + \lambda)} \\ 2\,ie^{kL2}\mu\,(k - kzero) & \frac{2\,i\left(L2\,k^2\lambda + L2\,k^2\mu - L2\,kkzero\,\lambda - kzero\,\mu\,L2\,k + k\lambda + 2\,k\mu - kzero\,\lambda - 2\,kzero\,\mu\right)\,\mu\,e^{kL2}}{k\,(\mu + \lambda)} \end{bmatrix}$$

```
> Bd2 := simplify(matrix([[coeff(eq7, a2), coeff(eq7, b2)], [coeff(eq8, a2), coeff(eq8, b2)]]));
```

$$Bd2 := \begin{bmatrix} 2\,e^{-kL2}\mu\,(k + kzero) & \frac{\left(2\,L2\,k^2\lambda + 2\,L2\,k^2\mu + 2\,L2\,kkzero\,\lambda + 2\,kzero\,\mu\,L2\,k - 2\,k\mu - 2\,kzero\,\mu\right)\,\mu\,e^{-kL}}{k\,(\mu + \lambda)} \\ \frac{-2\,i\,(k\lambda + 3\,k\mu + kzero\,\lambda + kzero\,\mu)\,\mu\,e^{-kL2}}{\lambda + 3\,\mu} & \frac{-2\,i\mu\,e^{-kL2}\left(L2\,k^2\lambda^2 + 4\,L2\,k^2\lambda\,\mu + 3\,L2\,k^2\mu^2 + L2\,kkzero\,\lambda^2 + 2\,kzero\,\mu\,L2\,k\lambda + kzero\,\mu^2 L2\,k - k\lambda^2 - 5\,k\lambda\,\mu - 6\,k\right.}{(\lambda + 3\,\mu)\,k\,(\mu + \lambda)} \end{bmatrix}$$

so we can get M_1 as

$$M_1 = Ad_1^{-1} * Ad_2 * Bd_2^{-1} * Bd_1$$

```
> Itd1 := evalm((inverse(Ad1)&* Ad2));
> Itd2 := evalm((inverse(Bd2)&* Bd1));
> L1 := delta; L2 := 0;
> Itd := simplify(evalm((Itd1&* Itd2)));
> ld := eigenvalues(Itd);
```

We can use simplified versions of the two DtN operator,

Appendix B

Appendix: Lemma Geneo

In this part, in order to keep this dissertation as self-contained as possible, we recall from [DJN15] some proof of lemmas used in chapter 4.

Symmetric Generalized Eigenvalue problem

Consider in a general case A and B as an abstract linear operators
Let V be a Hilbert space of dimension n endowed with a scalar product denoted by $(\cdot,\cdot)$. Let A and B be symmetric positive linear operators from V on V. We first assume that operator B is also definite. We introduce the following generalized eigenvalue problem

$$\boxed{\begin{aligned}&\text{Find } (\mathbf{y}_k, \mu_k) \in V \times \mathbb{R} \text{ such that}\\ &A\,\mathbf{y}_k = \mu_k\ B\,\mathbf{y}_k\,.\end{aligned}} \tag{B.1}$$

Since operator B is symmetric positive definite, the eigenvalues of (B.1) *can be chosen so that they form a both A-orthogonal and B-orthonormal basis of V:*

$$(B\mathbf{y}_k,\, \mathbf{y}_l) = \delta_{kl}, \quad (A\mathbf{y}_k,\, \mathbf{y}_l) = 0 \text{ for } k \neq l$$

where δ_{kl} is the classical Kroenecker symbol. We have the following result verified by the solutions of this problem

Lemma 32
Let $\tau > 0$ and define the space related to the eigenpairs of the problem (B.1)

$$Y_\tau := \mathit{Span}\left\{\mathbf{y}_k |\ \mu_k < \frac{1}{\tau}\right\}. \tag{B.2}$$

Let ξ_τ denote the projection on Y_τ parallel to Span$\{\mathbf{y}_k |\ \mu_k \geq \frac{1}{\tau}\}$.
Then, for all $\mathbf{y} \in V$ the following inequality holds

$$\boxed{(\,\mathbf{y} - \xi_\tau(\mathbf{y}), B(\mathbf{y} - \xi_\tau(\mathbf{y})\,) \leq \tau\,(A\mathbf{y}, \mathbf{y})\,.} \tag{B.3}$$

Proof. Let $\mathbf{y} \in V$, we have

$$\mathbf{y} = \sum_{k=1}^{n} (B\mathbf{y}, \mathbf{y}_k)\, \mathbf{y}_k = \underbrace{\sum_{\mu_k < \frac{1}{\tau}} (B\mathbf{y}, \mathbf{y}_k)\, \mathbf{y}_k}_{\in Y_\tau} + \underbrace{\sum_{\mu_k \geq \frac{1}{\tau}} (B\mathbf{y}, \mathbf{y}_k)\, \mathbf{y}_k}_{\in \text{Span}\{\mathbf{y}_k |\ \mu_k \geq \frac{1}{\tau}\}} .$$

Thus, we have:

$$\mathbf{y} - \xi_\tau(\mathbf{y}) = \sum_{\mu_k \geq \frac{1}{\tau}} (B\mathbf{y}, \mathbf{y}_k)\, \mathbf{y}_k \,,$$

so that using the B-orthonormality of the eigenvector basis,

$$(B(\mathbf{y} - \xi_\tau(\mathbf{y})),\, \mathbf{y} - \xi_\tau(\mathbf{y})) = \sum_{\mu_k \geq \frac{1}{\tau}} (B\mathbf{y}, \mathbf{y}_k)^2 \leq \tau \sum_{\mu_k \geq \frac{1}{\tau}} (B\mathbf{y}, \mathbf{y}_k)^2 \mu_k \,. \tag{B.4}$$

On the other hand, using the A-orthogonality of the eigenvector basis, we have:

$$\begin{aligned} (A\mathbf{y}, \mathbf{y}) &= \left(\sum_{k=1}^{n} (B\mathbf{y}, \mathbf{y}_k)\, A\mathbf{y}_k,\, \sum_{l=1}^{n} (B\mathbf{y}, \mathbf{y}_l)\, \mathbf{y}_l, \right) \\ &= \left(\sum_{k=1}^{n} (B\mathbf{y}, \mathbf{y}_k)\, A\mathbf{y}_k,\, \mathbf{y} \right) \\ &= \sum_{k=1}^{n} (B\mathbf{y}, \mathbf{y}_k)\, (A\mathbf{y}, \mathbf{y}_k) \\ &= \sum_{k=1}^{n} (B\mathbf{y}, \mathbf{y}_k)^2 \mu_k \geq \sum_{\mu_k \geq \frac{1}{\tau}} (B\mathbf{y}, \mathbf{y}_k)^2 \mu_k \,. \end{aligned} \tag{B.5}$$

Combining (B.4) and (B.5) ends the proof. ■

Other lemmas

The following lemmas are also used in chapter 4 in the proof of the continiuity and the stable decomposition lemmas

Lemma 33
Let M, $n, (n_i)_{1 \leq i \leq M}$ be positive integers, $Q_i \in \mathbb{R}^{n \times n_i}$, $1 \leq i \leq M$ be rectangular matrices and $A \in \mathbb{R}^{n \times n}$ be a symmetric positive definite matrix. Let $\widetilde{k_0}$ be an integer defined by:

$$\widetilde{k_0} := \max_{1 \leq j \leq M} \#\{i \,|\, Q_i^T A Q_j \neq 0\}.$$

Then for all $\mathbf{U}_i \in \mathbb{R}^{n_i}$, $1 \leq i \leq M$, we have the following estimate

$$\boxed{\left(\sum_{i=1}^{M} Q_i \mathbf{U}_i \right)^T A \left(\sum_{i=1}^{M} Q_i \mathbf{U}_i \right) \leq \widetilde{k_0} \sum_{i=1}^{M} \mathbf{U}_i^T \left(Q_i^T A Q_i \right) \mathbf{U}_i} \tag{B.6}$$

Proof. By Cauchy-Schwarz inequality, we have:

$$\begin{aligned}\left(\sum_{i=1}^{M} Q_i\mathbf{U}_i\right)^T A \left(\sum_{i=1}^{M} Q_i\mathbf{U}_i\right) &= \sum_{i,j|Q_i^T A\, Q_j\neq 0} \left(\mathbf{U}_i^T Q_i^T\right) A\, (Q_j\mathbf{U}_j) \\ &\leq \sum_{i,j|Q_i^T A\, Q_j\neq 0} \left(\mathbf{U}_i^T Q_i^T A\, Q_i\mathbf{U}_i\right)^{1/2} \left(\mathbf{U}_j^T Q_j^T A\, Q_j\mathbf{U}_j\right)^{1/2}.\end{aligned} \tag{B.7}$$

Let us introduce the connectivity matrix $C \in \mathbb{R}^M \times \mathbb{R}^M$ defined as follows:

$$C_{ij} = \begin{cases} 1 \text{ if } Q_i^T A\, Q_j \neq 0,\ 1 \leq i,j \leq M \\ 0 \text{ otherwise.} \end{cases} \tag{B.8}$$

and $\mathbf{v} \in \mathbb{R}^M$, $\mathbf{v} = (v_i)_{1\leq i\leq M}$ the vector of norms defined as

$$\mathbf{v} := \left((\mathbf{U}_1^T Q_1^T A Q_1\mathbf{U}_1)^{1/2}, \ldots, (\mathbf{U}_M^T Q_M^T A Q_M\mathbf{U}_M)^{1/2}\right)^T. \tag{B.9}$$

Note that we have

$$\|\mathbf{v}\|_2^2 = \sum_{i=1}^{M} v_i^2 = \sum_{i=1}^{M} \mathbf{U}_i^T (Q_i^T A Q_i)\mathbf{U}_i \tag{B.10}$$

and matrix C is symmetric as a consequence of the symmetry of A. With notations (B.8) and (B.9), estimate (B.7) becomes

$$\left(\sum_{i=1}^{M} Q_i\mathbf{U}_i\right)^T A \left(\sum_{i=1}^{M} Q_i\mathbf{U}_i\right) \leq \mathbf{v}^T C\mathbf{v}\,. \tag{B.11}$$

Note that we also have by definition of an operator norm:

$$\mathbf{v}^T C\mathbf{v} \leq \|C\|_2 \cdot \|v\|_2^2\,. \tag{B.12}$$

Since matrix C is symmetric, its 2-norm is given by the largest eigevalue in modulus, which is less than the infinity norm of C. It can be easily checked that $\|C\|_\infty = \widetilde{k_0}$ and by consequence we will have $\|C\|_2 \leq \widetilde{k_0}$. Finally, from estimates (B.10), (B.11) and (B.12), we have

$$\left(\sum_{i=1}^{M} Q_i\mathbf{U}_i\right)^T A \left(\sum_{i=1}^{M} Q_i\mathbf{U}_i\right) \leq \widetilde{k_0} \sum_{i=1}^{M} v_i^2 = \widetilde{k_0} \sum_{i=1}^{M} \mathbf{U}_i^T (Q_i^T A Q_i)\mathbf{U}_i$$

which ends the proof. ■

Lemma 34

Let k_1 be the maximal multiplicity of subdomains intersection, i.e. the largest integer m such that there exists m different subdomains whose intersection has a non zero measure.
Then, for all $\mathbf{U} \in \mathbb{R}^{\mathcal{N}}$, we have

$$\sum_{j=1}^{N} (R_j\mathbf{U})^T \widetilde{A}^j R_j\mathbf{U} \leq k_1\, \mathbf{U}^T A\mathbf{U} = k_1\, a(\mathbf{U},\mathbf{U}) \tag{B.13}$$

where matrices $\widetilde{A}^j$ is the local discretisation matrix.

Proof. This property makes use of the finite element discretisation framwork. Let $u_h := \sum_{k\in\mathcal{N}} U_k\phi_k$, then by definition

$$a_\Omega(u_h,u_h) = \mathbf{U}^T A\mathbf{U} \text{ and } a_{\Omega_j}(u_h,u_h) = (R_j\mathbf{U})^T\widetilde{A}^j R_j\mathbf{U}\,.$$

Since at most k_1 of the subdomains have a non zero measure intersection, the sum $\sum_{j=1}^{N} a_{\Omega_j}(u_h,u_h)$ cannot exceed k_1 times $a_\Omega(u_h,u_h)$. Let us be more explicit in the case of a Poisson equation. Inequality (B.13) reads:

$$\sum_{j=1}^{N}\int_{\Omega_j}|\nabla u_h|^2\,dx \;\leq k_1\int_{\Omega}|\nabla u_h|^2\,dx\,.$$

■

Bibliography

[ADLK01] P. R. Amestoy, I. S. Duff, J.-Y. L'Excellent, and J. Koster. "A fully asynchronous multifrontal solver using distributed dynamic scheduling". In: *SIAM J. Matrix Analysis and Applications* 23.1 (2001), pp. 15–41.

[ARSO14] A. M. Abdel-Rehim, A. Stathopoulos, and K. Orginos. "Extending the eigCG algorithm to nonsymmetric Lanczos for linear systems with multiple right-hand sides". In: *Numer. Linear Algebra Appl.* 21.4 (2014), pp. 473–493.

[BD97] J.-D. Benamou and B. Desprès. "A domain decomposition method for the Helmholtz equation and related optimal control problems". In: *J. Comput. Phys.* 136.1 (1997), pp. 68–82.

[BF91] F. Brezzi and M. Fortin. *Mixed and Hybrid Finite Element Methods*. Computational Mathematics. New York: Springer, 1991.

[BGL05] M. Benzi, G. H. Golub, and J. Liesen. "Numerical solution of saddle point problems". In: *Acta Numer.* 14 (2005), pp. 1–137.

[BS06] M. Benzi and V. Simoncini. "On the eigenvalues of a class of saddle point matrices". In: *Numer. Math.* 103.2 (2006), pp. 173–196.

[BS08] S. C. Brenner and L. R. Scott. *The Mathematical Theory of Finite Element Methods*. Applied Mathematics. New York: Springer, 2008.

[CB16] O. Colomés and S. Badia. "Segregated Runge-Kutta methods for the incompressible Navier-Stokes equations". In: *Internat. J. Numer. Methods Engrg.* 105.5 (2016), pp. 372–400.

[CDKN14] L. Conen, V. Dolean, R. Krause, and F. Nataf. "A coarse space for heterogeneous Helmholtz problems based on the Dirichlet-to-Neumann operator". In: *J. Comput. Appl. Math.* 271 (2014), pp. 83–99.

[CFS98] X.-C. Cai, C. Farhat, and M. Sarkis. "A minimum overlap restricted additive Schwarz preconditioner and applications to 3D flow simulations". In: *Contemporary Mathematics* 218 (1998), pp. 479–485.

[CK06] F. E. Cellier and E. Kofman. *Continuous system simulation*. Springer Science & Business Media, 2006.

[CS99] X.-C. Cai and M. Sarkis. "A restricted additive Schwarz preconditioner for general sparse linear systems". In: *SIAM Journal on Scientific Computing* 21 (1999), pp. 239–247.

[DGG09] V. Dolean, L. G. Giorda, and M. J. Gander. "Optimized Schwarz methods for Maxwell equations". In: *SIAM J. Scient. Comp.* 31.3 (2009), pp. 2193–2213.

[DJN15] V. Dolean, P. Jolivet, and F. Nataf. *An introduction to domain decomposition methods*. Algorithms, theory, and parallel implementation. Society for Industrial and Applied Mathematics (SIAM), Philadelphia, PA, 2015, pp. x+238.

[DJR92] B. Després, P. Joly, and J. E. Roberts. "A domain decomposition method for the harmonic Maxwell equations". In: *Iterative methods in linear algebra (Brussels, 1991)*. Amsterdam: North-Holland, 1992, pp. 475–484.

[DS99] E. De Sturler. "Truncation strategies for optimal Krylov subspace methods". In: *SIAM J. Numer. Anal.* 36.3 (1999), pp. 864–889.

[Des93] B. Després. "Domain decomposition method and the Helmholtz problem.II". In: *Second International Conference on Mathematical and Numerical Aspects of Wave Propagation (Newark, DE, 1993)*. Philadelphia, PA: SIAM, 1993, pp. 197–206.

[EG00] J. Erhel and F. Guyomarc'h. "An augmented conjugate gradient method for solving consecutive symmetric positive definite linear systems". In: *SIAM J. Matrix Anal. Appl.* 21.4 (2000), pp. 1279–1299.

[EG03] E. Efstathiou and M. J. Gander. "Why restricted additive Schwarz converges faster than additive Schwarz". In: *BIT* 43.suppl. (2003), pp. 945–959.

[EGLW12] Y. Efendiev, J. Galvis, R. Lazarov, and J. Willems. "Robust domain decomposition preconditioners for abstract symmetric positive definite bilinear forms". In: *ESAIM Math. Model. Numer. Anal.* 46.5 (2012), pp. 1175–1199.

[ESW14] H. C. Elman, D. J. Silvester, and A. J. Wathen. *Finite elements and fast iterative solvers: with applications in incompressible fluid dynamics*. Oxford University Press (UK), 2014.

[FD15] D. Forti and L. Dedè. "Semi-implicit BDF time discretization of the Navier-Stokes equations with VMS-LES modeling in a high performance computing framework". In: *Comput. & Fluids* 117 (2015), pp. 168–182.

[FML00] C. Farhat, A. Macedo, and M. Lesoinne. "A two-level domain decomposition method for the iterative solution of high-frequency exterior Helmholtz problems". In: *Numer. Math.* 85.2 (2000), pp. 283–303.

[GE10] J. Galvis and Y. Efendiev. "Domain decomposition preconditioners for multiscale flows in high contrast media: reduced dimension coarse spaces". In: *Multiscale Model. Simul.* 8.5 (2010), pp. 1621–1644.

[GHN01] M. J. Gander, L. Halpern, and F. Nataf. "Optimized Schwarz methods". In: *Domain decomposition methods in sciences and engineering (Chiba, 1999)*. DDM.org, Augsburg, 2001, 15–27 (electronic).

[GMN02] M. J. Gander, F. Magoulès, and F. Nataf. "Optimized Schwarz Methods without Overlap for the Helmholtz Equation". In: *SIAM J. Sci. Comput.* 24.1 (2002), pp. 38–60.

[GR12] V. Girault and P.-A. Raviart. *Finite element methods for Navier-Stokes equations: theory and algorithms*. Vol. 5. Springer Science & Business Media, 2012.

[GSV06] P. Gervasio, F. Saleri, and A. Veneziani. "Algebraic fractional-step schemes with spectral methods for the incompressible Navier-Stokes equations". In: *J. Comput. Phys.* 214.1 (2006), pp. 347–365.

[GW14] M. J. Gander and G. Wanner. "The Origins of the Alternating Schwarz Method". In: *Domain Decomposition Methods in Science and Engineering XXI*. Springer, 2014, pp. 487–495.

[Gan06] M. J. Gander. "Optimized Schwarz Methods". In: *SIAM J. Numer. Anal.* 44.2 (2006), pp. 699–731.

[HMNSXZ13] P. Havé, R. Masson, F. Nataf, M. Szydlarski, H. Xiang, and T. Zhao. "Algebraic domain decomposition methods for highly heterogeneous problems". In: *SIAM J. Sci. Comput.* 35.3 (2013), pp. C284–C302.

[HTJ88] T. Hagstrom, R. Tewarson, and A. Jazcilevich. "Numerical experiments on a domain decomposition algorithm for nonlinear elliptic boundary value problems". In: *Applied Mathematics Letters* 1.3 (1988), pp. 299 –302.

[Hec12] F. Hecht. "New development in FreeFem++". In: *J. Numer. Math.* 20.3-4 (2012), pp. 251–265.

[JHNP13] P. Jolivet, F. Hecht, F. Nataf, and C. Prud'homme. "Scalable Domain Decomposition Preconditioners For Heterogeneous Elliptic Problems". In: *Proceedings of the 2013 ACM/IEEE conference on Supercomputing*. SC13. Best paper finalist. ACM, 2013, 80:1–80:11.

[JM01] V. John and G. Matthies. "Higher-order finite element discretizations in a benchmark problem for incompressible flows". In: *International Journal for Numerical Methods in Fluids* 37.8 (2001), pp. 885–903.

[JN14] P. Jolivet and F. Nataf. *HPDDM: High-Performance Unified framework for Domain Decomposition methods, MPI-C++ library*. https://github.com/hpddm/hpddm. 2014.

[JNR98] C. Japhet, F. Nataf, and F.-X. Roux. "The Optimized Order 2 Method with a coarse grid preconditioner. Application to convection-diffusion problems". In: *Ninth International Conference on Domain Decompositon Methods in Science and Engineering*. Ed. by P. Bjorstad, M. Espedal, and D. Keyes. John Wiley & Sons, 1998, pp. 382–389.

[JT16] P. Jolivet and P.-H. Tournier. "Block Iterative Methods and Recycling for Improved Scalability of Linear Solvers". In: *Proceedings of the 2016 International Conference for High Performance Computing, Networking, Storage and Analysis*. SC16. IEEE. 2016.

[KF04] B. Koobus and C. Farhat. "A variational multiscale method for the large eddy simulation of compressible turbulent flows on unstructured meshes—application to vortex shedding". In: *Comput. Methods Appl. Mech. Engrg.* 193.15-16 (2004), pp. 1367–1383.

[KK98] G. Karypis and V. Kumar. *METIS: A Software Package for Partitioning Unstructured Graphs, Partitioning Meshes, and Computing Fill-Reducing Orderings of Sparse Matrices*. Tech. rep. http://glaros.dtc.umn.edu/gkhome/views/metis. Department of Computer Science, University of Minnesota, 1998.

[Key+13] D. E. Keyes et al. "Multiphysics simulations: challenges and opportunities". In: *International Journal of High Performance Computing Applications* 27.1 (2013), 4–83.

[LP08] J. Liesen and B. N. Parlett. “On nonsymmetric saddle point matrices that allow conjugate gradient iterations”. In: *Numer. Math.* 108.4 (2008), pp. 605–624.

[LSY98] R. B. Lehoucq, D. C. Sorensen, and C. Yang. *ARPACK users’ guide: solution of large-scale eigenvalue problems with implicitly restarted Arnoldi methods*. Vol. 6. SIAM, 1998.

[Lio89] P.-L. Lions. “On the Schwarz alternating method. II.” In: *Domain Decomposition Methods*. Ed. by T. Chan, R. Glowinski, J. Périaux, and O. Widlund. Philadelphia, PA: SIAM, 1989, pp. 47–70.

[Lio90] P.-L. Lions. “On the Schwarz alternating method. III: A variant for nonoverlapping subdomains”. In: *Third International Symposium on Domain Decomposition Methods for Partial Differential Equations , held in Houston, Texas, March 20-22, 1989*. Ed. by T. F. Chan, R. Glowinski, J. Périaux, and O. Widlund. Philadelphia, PA: SIAM, 1990.

[Man92] J. Mandel. “Balancing domain decomposition”. In: *Comm. on Applied Numerical Methods* 9 (1992), pp. 233–241.

[Mor02] R. B. Morgan. “GMRES with deflated restarting”. In: *SIAM J. Sci. Comput.* 24.1 (2002), 20–37 (electronic).

[NRS94] F. Nataf, F. Rogier, and E. de Sturler. *Optimal Interface Conditions for Domain Decomposition Methods*. Tech. rep. 301. CMAP (Ecole Polytechnique), 1994.

[NXDS11] F. Nataf, H. Xiang, V. Dolean, and N. Spillane. “A coarse space construction based on local Dirichlet to Neumann maps”. In: *SIAM J. Sci Comput.* 33.4 (2011), pp. 1623–1642.

[Nep91] S. V. Nepomnyaschikh. “Mesh theorems of traces, normalizations of function traces and their inversions”. In: *Sov. J. Numer. Anal. Math. Modeling* 6 (1991), pp. 1–25.

[Nep92] S. V. Nepomnyaschikh. “Decomposition and Fictious Domains for Elliptic Boundary Value Problems”. In: *Fifth International Symposium on Domain Decomposition Methods for Partial Differential Equations*. Ed. by D. E. Keyes, T. F. Chan, G. A. Meurant, J. S. Scroggs, and R. G. Voigt. Philadelphia, PA: SIAM, 1992, pp. 62–72.

[Nic87] R. A. Nicolaides. “Deflation of conjugate gradients with applications to boundary value problems”. In: *SIAM J. Numer. Anal.* 24.2 (1987), pp. 355–365.

[PSMJM06a] M. L. Parks, E. de Sturler, G. Mackey, D. D. Johnson, and S. Maiti. “Recycling Krylov subspaces for sequences of linear systems”. In: *SIAM J. Sci. Comput.* 28.5 (2006), 1651–1674 (electronic).

[PSMJM06b] M. L. Parks, E. de Sturler, G. Mackey, D. D. Johnson, and S. Maiti. “Recycling Krylov subspaces for sequences of linear systems”. In: *SIAM J. Sci. Comput.* 28.5 (2006), 1651–1674 (electronic).

[QV08] A. Quarteroni and A. Valli. *Numerical approximation of partial differential equations*. Vol. 23. Springer Science & Business Media, 2008.

[SCGT07a] A. St-Cyr, M. J. Gander, and S. J. Thomas. “Optimized multiplicative, additive, and restricted additive Schwarz preconditioning”. In: *SIAM J. Sci. Comput.* 29.6 (2007), 2402–2425 (electronic).

[SCGT07b] A. St-Cyr, M. J. Gander, and S. J. Thomas. "Optimized multiplicative, additive, and restricted additive Schwarz preconditioning". In: *SIAM J. Sci. Comput.* 29.6 (2007), 2402–2425 (electronic).

[SDHNPS14] N. Spillane, V. Dolean, P. Hauret, F. Nataf, C. Pechstein, and R. Scheichl. "Abstract robust coarse spaces for systems of PDEs via generalized eigenproblems in the overlaps". In: *Numer. Math.* 126.4 (2014), pp. 741–770.

[SDHNR13] N. Spillane, V. Dolean, P. Hauret, F. Nataf, and D. Rixen. "Solving generalized eigenvalue problems on the interfaces to build a robust two-level FETI method". In: *C. R. Math. Acad. Sci. Paris* 351.5-6 (2013), pp. 197–201.

[SO10] A. Stathopoulos and K. Orginos. "Computing and deflating eigenvalues while solving multiple right-hand side linear systems with an application to quantum chromodynamics". In: *SIAM J. Sci. Comput.* 32.1 (2010), pp. 439–462.

[STDKR96] M. Schäfer, S. Turek, F. Durst, E. Krause, and R. Rannacher. "Benchmark Computations of Laminar Flow Around a Cylinder". In: *Flow Simulation with High-Performance Computers II: DFG Priority Research Programme Results 1993–1995.* Wiesbaden: Vieweg+Teubner Verlag, 1996, pp. 547–566.

[SYEG00] Y. Saad, M. Yeung, J. Erhel, and F. Guyomarc'h. "A deflated version of the conjugate gradient algorithm". In: *SIAM J. Sci. Comput.* 21.5 (2000). Iterative methods for solving systems of algebraic equations (Copper Mountain, CO, 1998), pp. 1909–1926.

[Saa97] Y. Saad. "Analysis of augmented Krylov subspace methods". In: *SIAM J. Matrix Anal. Appl.* 18.2 (1997), pp. 435–449.

[Sag06] P. Sagaut. *Large Eddy Simulation for Incompressible Flows (Scientific Computation Series).* Springer-Verlag, 2006.

[Sch70] H. A. Schwarz. "Über einen Grenzübergang durch alternierendes Verfahren". In: *Vierteljahrsschrift der Naturforschenden Gesellschaft in Zürich* 15 (1870), pp. 272–286.

[Spi14] N. Spillane. "Robust domain decomposition methods for symmetric positive definite problems". Ph.D Thesis. Université Pierre et Marie Curie - Paris VI, Jan. 2014.

[TL15] X. Tu and J. Li. "A FETI-DP type domain decomposition algorithm for three-dimensional incompressible Stokes equations". In: *SIAM J. Numer. Anal.* 53.2 (2015), pp. 720–742.

[TNVE09] J. Tang, R. Nabben, C. Vuik, and Y. Erlangga. "Comparison of two-level preconditioners derived from deflation, domain decomposition and multigrid methods". In: *Journal of Scientific Computing* 39.3 (2009), pp. 340–370.

[TW05] A. Toselli and O. Widlund. *Domain Decomposition Methods - Algorithms and Theory.* Vol. 34. Springer Series in Computational Mathematics. Springer, 2005.

[UPDP11] A Uranga, P.-O. Persson, M Drela, and J Peraire. "Implicit large eddy simulation of transition to turbulence at low Reynolds numbers using a discontinuous Galerkin method". In: *International Journal for Numerical Methods in Engineering* 87.1-5 (2011), pp. 232–261.

[Vas08] P. S. Vassilevski. *Multilevel block factorization preconditioners.* Matrix-based analysis and algorithms for solving finite element equations. Springer, New York, 2008, pp. xiv+529.

[Nep91] S. Nepomnyaschikh. "Mesh theorems on traces, normalizations of function traces and their inversion." English. In: *Sov. J. Numer. Anal. Math. Model.* 6.3 (1991), pp. 223–242.

Espaces grossiers pour les méthodes de décomposition de domaine avec conditions d'interface optimisées

Abstract

The objective of this thesis is to design an efficient domain decomposition method to solve solid and fluid mechanical problems, for this, Optimized Schwarz methods (OSM) are considered and revisited. The optimized Schwarz methods were introduced by P.L. Lions. They consist in improving the classical Schwarz method by replacing the Dirichlet interface conditions by a Robin interface conditions and can be applied to both overlapping and non overlapping subdomains. Robin conditions provide us an another way to optimize these methods for better convergence and more robustness when dealing with mechanical problem with almost incompressibility nature. In this thesis, a new theoretical framework is introduced which consists in providing an Additive Schwarz method type theory for optimized Schwarz methods, e.g. Lions' algorithm. We define an adaptive coarse space for which the convergence rate is guaranteed regardless of the regularity of the coefficients of the problem. Then we give a formulation of a two-level preconditioner for the proposed method. A broad spectrum of applications will be covered, such as incompressible linear elasticity, incompressible Stokes problems and unstationary Navier-Stokes problem. Numerical results on a large-scale parallel experiments with thousands of processes are provided. They clearly show the effectiveness and the robustness of the proposed approach.

Keywords: *numerical analysis, domain decomposition methods, preconditioners, high performance computing, nearly-incompressible,geneo-2*

Résumé

L'objectif de cette thèse est la conception, l'analyse et l'implémentation d'une méthode de décomposition de domaine efficiente pour des problèmes de la mécanique des solides et des fluides. Pour cela les méthodes de Schwarz optimisée (OSM) sont considérées et révisées. Les méthodes de décomposition de domaine de Schwarz optimisées ont été introduites par P.L. Lions, elles apportent une amélioration aux méthodes de Schwarz classiques en substituant les conditions d'interface de Dirichlet par des conditions de type Robin et cela pour les méthodes avec ou sans recouvrement. Les conditions de Robin offrent un très bon levier qui nous permet d'aller vers l'optimalité des méthodes de Schwarz ainsi que la conception d'une méthode de décomposition de domaine robuste pour des problèmes de mécanique complexes comportant une nature presque incompressible. Dans cette thèse un nouveau cadre mathématique est introduit qui consiste à munir les méthodes de Schwarz optimisées (e.g. L'algorithme de Lions) d'une théorie semblable à celle déjà existante pour des méthodes de Schwarz additives, on définit un espace grossier pour lequel le taux de convergence de la méthode à deux niveaux peut être prescrit, indépendamment des éventuelles hétérogénéités du problème traité. Une formulation sous forme de preconditioneur de la méthode à deux niveaux est proposée qui permettra la simulation parallèle d'un large spectre de problèmes mécanique, tel que le problème d'élasticité presque incompressible, le problème de Stokes incompressible ainsi que le problème instationnaire de Navier-Stokes. Des résultats numériques issues de simulations parallèles à grande échelle sur plusieurs milliers de processeurs sont présentés afin de montrer la robustesse de l'approche proposée.

Mots clés : analyse numérique, méthodes de décomposition de domaines, preconditioneurs, calcul haute performance, presque-incompressible, geneo-2

Laboratoire Jacques-Louis Lions
4 place Jussieu – 75005 Paris – France